UNIVERSIDAD AUTONOMA DE AGUASCALIENTES

Centro de Ciencias Básicas

Departamento de Sistemas de Información

Maestría en Informática y Tecnologías Computacionales

Título:

Arquitectura para procesar imágenes de satélite para producir indicadores que apoyen en la medición de avances de los objetivos de desarrollo sostenible.

Caso práctico que presenta el **Ing. Armando Soto Valdez** para optar por el grado de: **Maestría en Informática y Tecnologías Computacionales**.

Comité tutoral:

- Tutor: ***Dr. Juan Muñoz López***
- Co-Tutor: ***Dra. María Dolores Torres Soto***
- Asesor: ***MC. Abel Alejandro Coronado Iruegas***

Aguascalientes, Ags. Enero de 2020

AUTORIZACIONES

UNIVERSIDAD AUTONOMA DE AGUASCALIENTES

M. en C. JORGE MARTÍN ALFÉREZ CHÁVEZ

DECANO DEL CENTRO DE CIENCIAS BÁSICAS

PRESENTE.

Como **tutor** designado del estudiante **ARMANDO SOTO VALDEZ** con ID **137105** quien realizó el trabajo de tesis titulado: **ARQUITECTURA PARA PROCESAR IMÁGENES DE SATÉLITE PARA PRODUCIR INDICADORES QUE APOYEN EN LA MEDICIÓN DE AVANCES DE LOS OBJETIVOS DE DESARROLLO SOSTENIBLE.**

Por medio del presente doy mi consentimiento de que la versión final del documento ha sido revisada y las correcciones se han incorporado apropiadamente, por lo que me permito emitir el voto aprobatorio para continuar con el procedimiento administrativo para la obtención del grado.

Aguascalienets, Ags., 08 de junio de 2020

Dr. Juan Muñoz Lopez

UNIVERSIDAD AUTONOMA DE AGUASCALIENTES

M. en C. JORGE MARTÍN ALFÉREZ CHÁVEZ

DECANO DEL CENTRO DE CIENCIAS BÁSICAS

PRESENTE.

Como **co-tutor** designado del estudiante **ARMANDO SOTO VALDEZ** con ID **137105** quien realizó el trabajo de tesis titulado: **ARQUITECTURA PARA PROCESAR IMÁGENES DE SATÉLITE PARA PRODUCIR INDICADORES QUE APOYEN EN LA MEDICIÓN DE AVANCES DE LOS OBJETIVOS DE DESARROLLO SOSTENIBLE.**

Por medio del presente doy mi consentimiento de que la versión final del documento ha sido revisada y las correcciones se han incorporado apropiadamente, por lo que me permito emitir el voto aprobatorio para continuar con el procedimiento administrativo para la obtención del grado.

Aguascalienets, Ags., 08 de junio de 2020

Dra. María Dolores Torres Soto

UNIVERSIDAD AUTÓNOMA DE AGUASCALIENTES

M. en C. JORGE MARTÍN ALFÉREZ CHÁVEZ

DECANO DEL CENTRO DE CIENCIAS BÁSICAS

PRESENTE.

Como **lector** designado del estudiante **ARMANDO SOTO VALDEZ** con ID **137105** quien realizó el trabajo de tesis titulado: **ARQUITECTURA PARA PROCESAR IMÁGENES DE SATÉLITE PARA PRODUCIR INDICADORES QUE APOYEN EN LA MEDICIÓN DE AVANCES DE LOS OBJETIVOS DE DESARROLLO SOSTENIBLE**.

Por medio del presente doy mi consentimiento de que la versión final del documento ha sido revisada y las correcciones se han incorporado apropiadamente, por lo que me permito emitir el voto aprobatorio para continuar con el procedimiento administrativo para la obtención del grado.

Aguascalientes, Ags., 08 de junio de 2020

MC. Abel Alejandro Coronado Iruegas

DICTAMEN DE LIBERACION ACADEMICA PARA INICIAR LOS TRAMITES DEL EXAMEN DE GRADO

Fecha de dictaminación dd/mm/aa: 22/06/20

NOMBRE: Armando Soto Valdez **ID** 137105

PROGRAMA: Maestría en Informática y Tecnologías Computacionales **LGAC (del posgrado):** La Ingeniería/Enfoque de sistemas para el mejoramiento de los procesos organizacionales usando SI/TI

TIPO DE TRABAJO: ( ) Tesis (X) Trabajo práctico

TITULO: Arquitectura para procesar imágenes de satélite para producir indicadores que apoyen en la medición de avances de los ODS.

IMPACTO SOCIAL (señalar el impacto logrado): Académico

INDICAR SI/NO SEGÚN CORRESPONDA:

Elementos para la revisión académica del trabajo de tesis o trabajo práctico:

SI	El trabajo es congruente con las LGAC del programa de posgrado
SI	La problemática fue abordada desde un enfoque multidisciplinario
SI	Existe coherencia, continuidad y orden lógico del tema central con cada apartado
SI	Los resultados del trabajo dan respuesta a las preguntas de investigación o a la problemática que aborda
SI	Los resultados presentados en el trabajo son de gran relevancia científica, tecnologíca o profesional según el área
SI	El trabajo demuestra más de una aportación original al conocimiento de su área
NO	Las aportaciones responden a los problemas prioritarios del país
SI	Generó transferecia del conocimiento o tecnológica
SI	Cumpe con la ética para la investigación (reporte de la herramienta antiplagio)

El egresado cumple con lo siguiente:

SI	Cumple con lo señalado por el Reglamento General de Docencia
SI	Cumple con los requisitos señalados en el plan de estudios (créditos curriculares, optativos, actividades complementarias,estancia, predoctoral, etc)
SI	Cuenta con los votos aprobatorios del comité tutoral, en caso de los posgrados profesionales si tiene solo tutorpodrá liberar solo el tutor
SI	Cuenta con la carta de satisfacción del Usuario
SI	Coincide con el tituo y objetivo registrado
SI	Tiene congruencia con cuerpos académicos
SI	Tiene el CVU del Conacyt actualizado
	Tiene el artículo aceptado o publicado y cumple con los requisitos institucionales (en caso que proceda)

En caso de Tesis por artículos científicos publicados

	Aceptación o Publicación de los articulos según el nivel del programa
	El estudiante es el primer autor
	El autor de correspondencia es el Tutor del Núcleo Académico Básico
	En los artículos se ven reflejados los objetivos de la tesis, ya que son producto de este trabao de investigación.
	Los artículos integran los capítulos de la tesis y se presentan en el idioma en que fueron publicados
	La aceptación o publicación de los artículos en revistas indexadas de alto impacto

Con base a estos criterios, se autoriza se continúen con los trámites de titulación y programación del examen de grado

Sí X

No

FIRMAS

Elaboró:

* NOMBRE Y FIRMA DEL CONSEJERO SEGÚN LA LGAC DE ADSCRIPCION: DR. JOSÉ MANUEL MORA TAVAREZ

NOMBRE Y FIRMA DEL SECRETARIO TÉCNICO: M.I.T.C. JORGE EDUARDO MACÍAS LUÉVANO

*** En caso de conflicto de intereses, firmará un revisor miembro del NAB de la LGAC correspondiente distinto al tutor o miembro del comité tutoral, asignado por el Decano**

Revisó:

NOMBRE Y FIRMA DEL SECRETARIO DE INVESTIGACIÓN Y POSGRADO: DRA. NAYDEE MARTÍNEZ BUVALCABA

Autorizó:

NOMBRE Y FIRMA DEL DECANO: M.C. JORGE MARTÍN ALFÉREZ CHÁVEZ

Nota: procede el trámite para el Depto. de Apoyo al Posgrado

En cumplimiento con el Art. 105C del Reglamento General de Docencia que a la letra señala entre las funciones del Consejo Académico: .... Cuidar la eficiencia terminal del programa de posgrado y el Art. 105F las funciones del Secretario Técnico, llevar el seguimiento de los alumnos.

Elaborado por: D. Apoyo al Posg.
Revisado por: D. Control Escolar/D. Gestión de Calidad.
Aprobado por: D. Control Escolar/ D. Apoyo al Posg.

Código: DO-SEE-FO-15
Actualización: 01
Emisión: 28/04/20

AGRADECIMIENTOS

Quiero expresar mi agradecimiento al Instituto Nacional de Estadística y Geografía por darme la oportunidad de realizar mi caso de estudio práctico en sus instalaciones, además de financiar el equipo requerido para alcanzar los objetivos establecidos en el desarrollo del proyecto

De igual forma quiero agradecer al núcleo académico de la Maestría en Informática y Tecnologías Computacionales, de la Universidad Autónoma de Aguascalientes; por el apoyo brindado, las experiencias compartidas y el conocimiento adquirido a lo largo de este tiempo.

De manera especial quiero externar mi más sincero agradecimiento al Dr. Juan Muñoz López, al MC. Abel Alejandro Coronado Iruegas y al MC. Edgar Oswaldo Diaz; por acompañarme durante el proceso para la realización de este trabajo.

Por último, pero no menos importante; agradezco el eterno apoyo recibido por parte de mi familia, quienes siempre me han motivado a alcanzar mis sueños y vivirlos.

¡Gracias!

DEDICATORIAS

A mi compañera de escuela y de trabajo, a mi mejor amiga, a mi novia y a mi futura esposa. Te amo Diana García.

A mis padres, quienes siempre han estado presentes en mis logros y me motivan a ser mejor cada día.

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN EN INGLÉS (ABSTRACT):

This document provides an explanation of the process that was carried out to design a hardware architecture with the ability to process and store satellite images. In addition, it offers an approach to a new research tool developed in Australia, the data cube; a powerful software that is capable of performing spatial and temporal analysis of Earth's observations, obtaining the power to generate extremely useful products for official statistical offices and that can support the measurement of the progress of sustainable development goals.

RESUMEN EN ESPAÑOL:

Este documento provee una explicación del proceso que se realizó para diseñar una arquitectura de hardware con la capacidad de poder procesar y almacenar imágenes de satélite. Además, ofrece un acercamiento a una nueva herramienta de investigación desarrollada en Australia, el cubo de datos; un software potente que es capaz de realizar análisis espacial y temporal sobre las observaciones de la Tierra, consiguiendo el poder de generar productos sumamente útiles para las oficinas de estadística oficial y que pueden apoyar en la medición de los avances de los objetivos de desarrollo sostenible.

1. INTRODUCCIÓN

El caso práctico que se presenta mediante este documento muestra el reto que se tuvo en INEGI (Instituto Nacional de Estadística y Geografía) para poder cumplir con el compromiso internacional, ante la Organización de las Naciones Unidas (ONU), de medir los indicadores de avance de los Objetivos de Desarrollo Sostenible (ODS) para México; descrito en el capítulo 2 del documento, donde se expone la problemática a resolver.

En INEGI se planteó la posibilidad de diseñar y crear una estructura tecnológica capaz de apoyar la medición de los indicadores, utilizando imágenes de satélite y herramientas capaces de procesarlas con el fin de obtener estadísticas oficiales acerca del avance de los mencionados ODS.

2. Planteamiento de la Problemática a Atender

El INEGI es un organismo público autónomo responsable de captar y difundir información estadística y geográfica del territorio, los recursos, la población y la economía de México; y apoyar en la toma de decisiones en los diferentes sectores de la sociedad. (INEGI, 2018)

Para la obtención de información geográfica sobre el relieve, la vegetación, el clima, suelo, agua, localidades, entre otros temas, el INEGI hace uso de imágenes de percepción remota; tales como las fotografías aéreas, orto fotos digitales e imágenes de satélite (INEGI, 2018). Esta última fuente de información será la base del caso de estudio que describe este documento.

Actualmente el INEGI tiene el compromiso con la ONU de reportar los resultados de las mediciones sobre los ODS del país cada cierto periodo de tiempo; sin embargo, es importante mencionar que al INEGI no le corresponde medir todos los rubros que componen a los indicadores de medición, sino que solo el de unos cuantos, definidos por el comité técnico especializado de los ODS del Sistema Nacional de Información Estadística y Geográfica (SNIEG), quien específica a cual institución de gobierno le corresponde la medición de cada uno de los indicadores de los ODS; no obstante, el INEGI es quien recopila toda la información de las otras instituciones públicas y las reporta ante la ONU.

Durante los últimos meses el INEGI se ha propuesto implementar una tecnología que apoye con la medición de ciertos indicadores de los ODS, para lo cual se planea aprovechar los avances logrados en cuanto al manejo de las imágenes satelitales y con ello obtener resultados más oportunos y a un menor costo. El problema que se presenta al tratar con imágenes de este tipo radica en que se necesita un alto poder de procesamiento y la contemplación del almacenamiento para grandes volúmenes de información, ya que son archivos cuyo tamaño oscila alrededor de los 7.5 TB para un año de imágenes del país.

Debido a lo anterior, el INEGI ha decidido empezar a trabajar con el diseño y la construcción de una arquitectura tecnológica que soporte los requerimientos para trabajar sin problemas con imágenes satelitales, de manera que facilite el procesamiento de estas y lograr con ello la puesta a disposición de diferentes productos que apoyen a las tareas de las diferentes áreas del instituto.

La elaboración y construcción de una arquitectura especial para el procesamiento de información geoespacial, no solo beneficiará al área encargada del monitoreo de los ODS, sino que también puede ser aprovechada por otros proyectos institucionales que requieran del uso de información de imágenes satelitales, o incluso podría ser la base para el ofrecimiento de diferentes servicios públicos que el INEGI puede ofrecer a la población en general.

3. OBJETIVOS

Los objetivos para conseguir de esta intervención de caso práctico en el INEGI son los que se muestran a continuación:

1. Diseñar un modelo arquitectónico que permita el manejo y análisis de imágenes satelitales.

2. Procesar información en imágenes de satélite para generar indicadores de medición y avance de los ODS.

4. Fundamentación Teórica

4.1 Sistemas de observación de la Tierra.

La percepción remota es una tecnología que es aplicada en la observación y estudio de los sistemas terrestres, los datos obtenidos para esta tecnología provienen principalmente de imágenes de satélite con el propósito de estudiar la superficie terrestre, los océanos y la atmósfera desde el espacio. (United Nations, y otros, 2017)

Hay que tener en cuenta que la humanidad afecta de manera directa o indirecta la apariencia de la Tierra; lo podemos ver con la reducción de las áreas verdes provocada por la tala inmoderada de árboles, por el crecimiento de las grandes ciudades, o incluso por la agricultura que se lleva a cabo en determinadas zonas; por lo que podemos concluir que las actividades socioeconómicas también pueden ser analizadas y estudiadas desde el espacio. (United Nations, y otros, 2017)

Los resultados de los estudios aplicados a la recolección de datos socioeconómicos apoyan a las instituciones gubernamentales con el cálculo y la medición de estadísticas oficiales, enfocándose en medir los efectos de la vida social en cuanto a su evolución, su distribución, sus impactos, su sustentabilidad, entre otros rubros (United Nations, y otros, 2017). En México, la institución dedicada a la medición y publicación de las estadísticas oficiales es el INEGI. (INEGI, 2018)

4.2 Sensores activos y pasivos.

Los datos que se extraen de las imágenes satelitales sobre algún objeto; valor, estado o condición, se obtienen a partir de una señal electromagnética que es emitida o reflejada por el objeto y que es captada por el satélite desde el espacio. Los satélites logran la captación de las ondas electromagnéticas por medio de sensores; estos sensores se clasifican en activos y pasivos según su función, como se muestra en la Figura 1. (Richards & Jia, 2006)

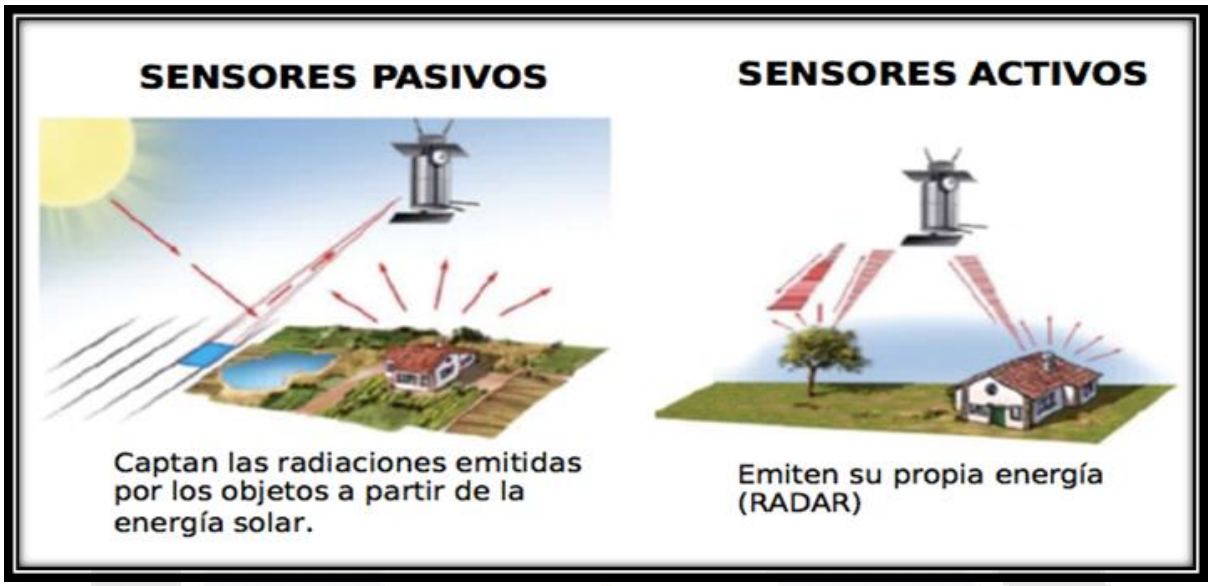


Figura 1. Funcionamiento de los sensores activos/pasivos.

(EcuRed: https://www.ecured.cu/Sensor_remoto)

Los sensores pasivos son aquellos que detectan las señales electromagnéticas naturales, por ejemplo, la radiación del sol o la radiación térmica emitida por algún objeto en particular. Mientras que los sensores activos proporcionan su propia fuente de radiación sobre los objetos, los cuales al iluminarse regresan señales que detecta el sensor. (Richards & Jia, 2006)

Las ondas electromagnéticas captadas por un sensor pasivo pueden variar entre las que son visibles para el ojo humano y las que no. Las longitudes de onda visible son las correspondientes a la escala de colores azul, verde, amarilla, naranja y roja; mientras que las longitudes de onda invisibles son las de infrarrojo cercano (NIR, Near-Infrared), las de infrarrojo de onda corta (SWIR, Short Wave Infrared), las longitudes de onda de infrarrojo medio (MIR, Mid Infrared) y, aunque aún se encuentran en investigación, las longitudes de onda de infrarrojo térmico (TIR, Thermal Infrared); utilizadas para la medición de temperaturas en la región de cobertura. (United Nations, y otros, 2017)

"Los sensores VIS, NIR, SWIR y MIR operan principalmente durante el día, con algunas excepciones, como cuando se desea medir el uso de luces nocturnas para detectar fuentes de luz marinas o terrestres". Los sensores TIR también pueden detectar emisiones por la noche y usarse para estimar la humedad del suelo en la tierra o la salinidad del océano. (United Nations, y otros, 2017)

4.3 Categorías de las imágenes de satélite.

Existen 3 categorías de imágenes satelitales, identificadas por su resolución espacial y su resolución temporal: (Martinez & Calvo, 2019)

- Imágenes de Baja Orbita de la Tierra (LEO, Low Earth polar Orbiting)
- Imágenes de Media Orbita de la Tierra (MEO, Medium Earth Orbit)
- Imágenes de un satélite Geoestacionario (GEO, Geostationary)

Los satélites LEO, por lo general flotan entre los 400 y 800 kilómetros siguiendo una trayectoria que cruza por los dos polos de la Tierra a una velocidad promedio de 28,000 km/h, por lo que tendrían la capacidad de recorrer una vuelta al planeta en aproximadamente 90 minutos; logrando así generar una imagen global cada cierto periodo de tiempo, que, dependiendo del tipo de satélite, puede tomar entre un día y un mes. Debido a la corta distancia que tienen de la Tierra, su resolución espacial puede ser tan alta como 30 cm por píxel en imágenes en blanco y negro, o aproximadamente de 1 m por píxel en imágenes a color o con bandas multiespectrales, ambas disponibles de manera comercial. (Martinez & Calvo, 2019)

Los satélites más conocidos que están dentro de la categoría LEO y que están disponibles de manera gratuita son: (United Nations, y otros, 2017)

- **MODIS**, desde 1999 y con una resolución espacial de 250/500/1000 m por píxel.

- **Landsat**, desde 1973 y con una resolución inicial de 80 m por píxel, de 1984 a la actualidad, tiene una resolución de 30 m por pixel.

- **Sentinel**, desde 2015 y con una resolución de 10 m por píxel.

Figura 2. Imagen MODIS tomada sobre la costa del Golfo en Estados Unidos.

(**NASA**:https://modis.gsfc.nasa.gov/gallery/individual.php?db_date=2020-01-08)

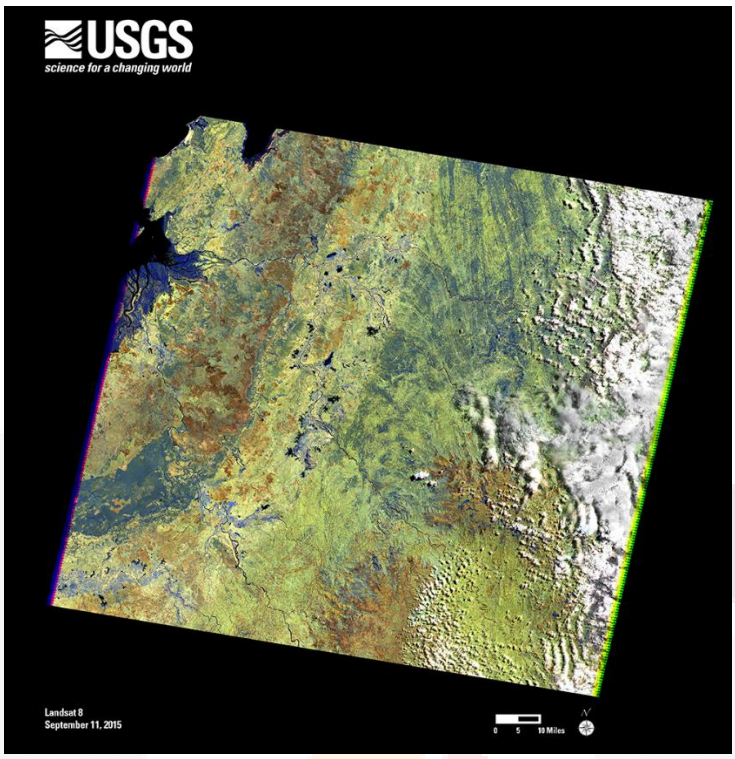


Figura 3. Muestra de imagen Landsat 8 capturada sobre el norte de Madagascar.

(**NASA**: https://landsat.gsfc.nasa.gov/landsat-7-captures-two-millionth-scene/)

Figura 4. Muestra de imagen Sentinel 2A capturada sobre Brindisi, Italia.

(**ESA**: https://www.esa.int/ESA_Multimedia/Images/2017/03/Brindisi_Italy)

Los satélites de la categoría MEO, se encuentran a aproximadamente 20,000 km de distancia de la Tierra y están enfocados principalmente en el ámbito de las comunicaciones y de la navegación, así como también son utilizados con fines de conocimiento para la ciencia; según la Administración Nacional de la Aeronáutica y del Espacio (NASA), este tipo de satélites recorren una órbita en aproximadamente entre 12 y 20 horas. (Martinez & Calvo, 2019)

En cuanto a los satélites de la categoría GEO, están localizados sobre el ecuador a una distancia de 36,000 km de la Tierra, donde pueden permanecer en el mismo lugar. Originalmente fueron creados con fines de uso meteorológico, sin embargo, se han estado volviendo tan sofisticados que ya no solo los usan para ese fin, sino que también son útiles para las telecomunicaciones. (Martinez & Calvo, 2019)

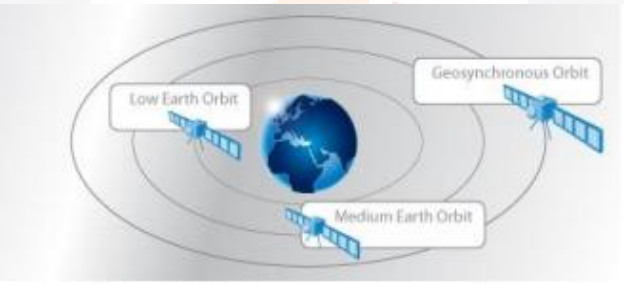


Figura 5. Representación visual de las órbitas de los satélites.

(ONU, 2017)

4.4 Big Data en la integración de datos geoespaciales.

Algo que es importante considerar a la hora de trabajar con la información proveniente de las imágenes de satélite, es que regularmente se tiene que involucrar el uso de tecnologías con la capacidad necesaria para el almacenamiento y procesamiento de grandes volúmenes de información; por ejemplo, la información de un año de imágenes de satélite Landsat 7 sobre la superficie del país (México) ocupan cerca de 7.5 TB, por lo que para pensar en tener la historia de México en imágenes satelitales se requiere contar con un gran espacio de almacenamiento. (INEGI, 2019)

El hecho de involucrar el uso de grandes volúmenes de información para cualquier actividad nos introduce al concepto de Big data; que es definido por la UNECE como: (UNECE, 2016)

Big Data: *"Datos que son difíciles de recolectar, almacenar o procesar con las aplicaciones de datos tradicionales"*.

En la actualidad, el sector privado hace un uso amplio de Big Data para su fortalecimiento comercial; un ejemplo que podemos ver claramente es el uso de Big Data dentro de las redes sociales, que además de contar con el almacenamiento suficiente para esas grandes fuentes de información, cuentan con la capacidad de procesamiento para el análisis estadístico de esos datos. (Lohr, 2012)

Los datos se han vuelto cada vez más accesibles, al mismo tiempo que más comprensibles para las computadoras. La mayor parte del Big Data está compuesta por palabras, imágenes y videos obtenidos de diferentes fuentes; por lo que tenemos como resultado un conjunto de datos no estructurados que no pueden ser administrados por las bases de datos tradicionales. (Lohr, 2012)

Hablar de datos geoespaciales es referirse siempre a Big Data; los datos geoespaciales generalmente se refieren a conjunto de datos espaciales que exceden la capacidad de los sistemas informáticos actuales. (Lee & Kang, 2015)

Para el año 2009, los datos de ubicación personal estaban cerca de 1 PB por año, creciendo alrededor de 20% anual, según el Instituto Global McKinsey. Sin embargo, para el año 2015 la información Geoespacial Global generó 2.5 quintillones de bytes de datos todos los días, según la ONU. Además, como dato adicional, en Google se generaron alrededor de 25 PB de datos por día, y una parte significativa de los datos cae en el ámbito de los datos espaciotemporales. Todo lo anterior probablemente ocasionado por el incremento en el uso de dispositivos móviles. (Lee & Kang, 2015)

Un grupo de investigadores de las universidades de Oxford y Nottingham refieren a que el termino de Big Data puede dividirse en 2 categorías; los datos generados por humanos y los datos generados por máquinas. Los datos generados por sensor, como los satélites, entran en la categoría de datos generados por máquina y son utilizados para fines de investigación desde la década de los 90. (Gartner, Huang, Research Group Cartography, & Vienna University of Technology, 2015)

El análisis y procesamiento manual de las imágenes de satélite es algo con lo que se está trabajando desde hace ya varios años; sin embargo, el análisis automático mediante nuevos modelos estadísticos, que permiten analizar los datos generados por máquina, es un tema que está tomando mucha relevancia en el mundo del Big Data actualmente. (Gartner, Huang, Research Group Cartography, & Vienna University of Technology, 2015)

Con el avance de la tecnología que se ha desarrollado durante los últimos años, se han podido crear nuevos modelos de computación con los que se puede almacenar, procesar y analizar grandes volúmenes de datos; en adición a esto, es importante destacar los avances del procesamiento paralelo y el computo distribuido, que permiten una mejora significativa en los tiempos de espera de resultados. (Gartner, Huang, Research Group Cartography, & Vienna University of Technology, 2015)

Los sistemas de Big Data geoespaciales requieren de ciertos tipos de técnicas, algoritmos para tener una gestión eficiente, análisis y uso compartido de información; estas necesidades representan los desafíos que tiene la Geo-Computación. De hecho, el trabajar con datos geoespaciales demuestra los límites de los sistemas de información y los marcos computacionales; es por eso que varios autores refieren que trabajar con este tipo de datos puede representar uno de los mayores desafíos de Big Data. (Gartner, Huang, Research Group Cartography, & Vienna University of Technology, 2015)

4.5 Algoritmos de análisis para imágenes de satélite.

Existen algoritmos que nos permiten analizar las imágenes de satélite para con ello poder obtener la distribución espacial de las variables; como color, profundidad, concentración, densidad, masa, entre otros. Además, es posible conocer la condición de alguna región especifica añadiendo como factor el aspecto temporal repetidamente en esa zona, llegando a producir algunas estadísticas oficiales. (United Nations, y otros, 2017)

La naturaleza de estos algoritmos permite clasificarlos en 5 categorías conforme a los métodos que implementan; en donde las primeras 3 son acerca de métodos que están basados en el análisis de pixeles, el cuarto toma la información de los pixeles, así como la información espacial y contextual de la zona, por último, el quinto método es un algoritmo hibrido, es decir, puede combinar los métodos anteriormente descritos. A continuación, se lista cada uno de los métodos y una breve explicación: (United Nations, y otros, 2017)

1. **Métodos empíricos:** Se establece una relación estadística entre las bandas espectrales utilizadas y la variable medida, basada en la superficie terrestre. Este método es el menos adecuado para el análisis automatizado en grandes áreas y condiciones variable; tales como: ángulo solar, estación, latitud, pendiente y aspecto del terreno, condiciones atmosféricas, entre otras; a menos que esté acompañado de una actividad de medición de campo significativa y continua, preferiblemente con cada paso elevado de satélite.

2. **Métodos semi empíricos:** Se establece una relación causal entre las bandas espectrales utilizadas y la variable evaluada. Este método es menos propenso a proporcionar resultados que no sean ciertos, sin embargo, puede resultar un error fuera del rango basado en el campo. Este método tiene una idoneidad media para el análisis automatizado en

grandes áreas y tiene requisitos menos estrictos para el muestreo frecuente en el campo, como fue el caso del punto anterior.

3. **Métodos de inversión basados en la física:** Este método también es conocido como método de inversión semianalítico. En este método, todas las variables requeridas se evalúan en una inversión espectral simultáneamente, que proporciona una consistencia basada en la física de los resultados y es más adecuado para el análisis automatizado en grandes áreas; siempre que el modelo de inversión esté correctamente parametrizado para las variables deseadas.

4. **Método de análisis de imagen basado en objetos (OBIA):** Este método combina información espacial, de patrones, de textura y espectral con información contextual supervisada por un operador humano. Este método tiene idoneidad para el análisis automatizado en todo el sistema para el que fue desarrollado, cada vez más utilizado.

5. **Inteligencia artificial y métodos de aprendizaje automático:** Estos son modelos estadísticos inteligentes para identificar relaciones altamente complejas en los datos. Dependiendo de cómo se entrenan estos métodos; utilizando mediciones de campo o modelos basados en la física, cruzan los ámbitos de los métodos de inversión empíricos a semi empíricos a físicos y OBIA.

4.6 Las imágenes de satélite y las estadísticas oficiales.

La Asamblea General y el Consejo Económico y Social de las Naciones Unidas resaltan la importancia fundamental de las estadísticas oficiales, teniendo presente la función decisiva que desempeña esta información de alta calidad para el análisis y la adopción de decisiones normativas bien fundadas en apoyo del desarrollo sostenible, la

paz y la seguridad, así como para el conocimiento mutuo y el comercio entre los Estados y los pueblos. (INEGI, 2019)

Considerando también que la confianza del público en integridad de los sistemas estadísticos oficiales y la credibilidad en las estadísticas dependen del respeto de los valores y principios fundamentales de las estadísticas oficiales, los cuales son cruciales al momento de la rendición de cuentas de los organismos de estadística de cada país, en el caso de México es el INEGI. Los principios fundamentales de las estadísticas oficiales, según la asamblea general de la ONU, se listan a continuación: (INEGI, 2019)

1. **Relevancia, imparcialidad y acceso equitativo:** Las estadísticas oficiales constituyen un elemento indispensable en un sistema de información que proporcionan, al gobierno y al público en general, datos acerca de la situación económica, demográfica, social y ambiental de un país.

2. **Patrones profesionales, principios científicos y ética:** Para mantener la confianza en las estadísticas oficiales, los organismos de estadística deben ser estrictamente profesionales e incluir principios científicos y de ética profesional sobre los métodos y procedimientos para la reunión, el procesamiento, el almacenamiento y la presentación de los datos estadísticos.

3. **Responsabilidad y transparencia:** Para facilitar una interpretación correcta de los datos, los organismos de estadística deben presentar información conforme a normas científicas sobre las fuentes, métodos y procedimientos de la estadística.

4. **Prevención del mal uso:** Los organismos de estadística tienen derecho a formular observaciones sobre interpretaciones erróneas y la utilización indebida de las estadísticas.

5. **Fuentes de estadísticas oficiales:** Los datos para fines estadísticos pueden obtenerse de todo tipo de fuentes, ya sea encuestas o registros administrativos. Los organismos de estadística han de seleccionar la fuente con respecto a la calidad, la oportunidad, el costo y la carga que impondrá a los encuestados.

6. **Confidencialidad:** Los datos individuales que reúnan los organismos de estadística, que se refieran a personas naturales o jurídicas, deben ser estrictamente confidenciales y utilizarse exclusivamente para fines estadísticos.

7. **Legislación:** Se han de dar a conocer al público las leyes, reglamentos y medidas que rigen la operación de los sistemas estadísticos.

8. **Coordinación nacional:** La coordinación entre los organismos de estadística a nivel nacional es indispensable para lograr la coherencia y eficiencia del sistema estadístico.

9. **Uso de patrones internacionales:** La utilización de cada país acerca de conceptos, clasificaciones y métodos internacionales fomenta la coherencia y eficiencia de los sistemas estadísticos a nivel oficial.

10. **Cooperación internacional:** La cooperación bilateral y multilateral en la esfera de la estadística contribuye a mejorar los sistemas de estadística y geografía en todos los países.

Las imágenes de satélite son fuentes de información que pueden ser explotadas para apoyar al INEGI en el cálculo de estadísticas oficiales acerca de diferentes rubros; tales como el índice de vegetación, análisis de cuerpos de agua, índice de crecimiento urbano, entre otros. Además, son tomadas como soporte para otros productos de INEGI, como puede ser la “Carta de uso de suelo y vegetación”, la clasificación de cultivos a

nivel estado, la generación de geomedianas nacionales y la recolección de información para el registro de “Objetivos de Desarrollo Sostenible”. (INEGI, 2019)

4.7 Aplicación y contribuciones del uso de imágenes de satélite.

La teledetección desde plataformas espaciales proporciona datos valiosos para la generación de mapas, monitoreo ambiental, análisis y gestión de desastres e inteligencia civil y militar. Sin embargo, para explorar completamente este conjunto de datos se debe extraer la información apropiada para importarla a los sistemas de geo-información y así permitir procesos de decisión eficientes. (Anderson, Ryan, Sonntag, Kavvada, & Friedl, 2017)

El enfoque orientado a objetos que nos pueden ofrecer distintas herramientas SIG (Sistemas de Información Geográfica; también conocidas como GIS, por sus siglas en inglés), puede contribuir a un potente análisis automático y/o semiautomático para la mayoría de las aplicaciones de teledetección de la Tierra; puesto que el uso de los métodos de procesamiento de señales o aquellos que son basados en píxeles exploran los ricos contenidos de información provenientes de los sensores remotos. La combinación de los métodos orientados a objetos con los métodos difusos permite implementar el conocimiento experto y representativo para el flujo de trabajo de las imágenes de satélite en los sistemas SIG. (Benz, Hofmann, Willhauck, Lingenfelder, & Heynen, 2004)

En las siguientes páginas del documento se platicarán algunos de los casos de uso que países como Reino Unido, México e India aplican sobre las imágenes de satélite para cumplir objetivos específicos que apoyan a las estadísticas oficiales de cada país.

4.7.1 Caso Reino Unido.

En el Reino Unido, investigaciones sugieren que los datos provenientes de las imágenes de satélite en combinación con otras fuentes de datos geoespaciales tienen un gran potencial para apoyar en la generación de nuevas políticas gubernamentales y servicios proporcionados por el mismo gobierno. En promedio, el gobierno de este país

gasta 175 millones de euros en la obtención, análisis y procesamiento de imágenes de satélite; a través de la Agencia Espacial del Reino Unido (UKSA) y en colaboración con la Agencia Espacial Europea (ESA). (London Economics, 2018)

La industria en el Reino Unido tiene la oportunidad de capitalizar el uso de imágenes de satélite, aprovechando las capacidades nacionales y la base de investigación existente, para lograr ser líder en este mercado en crecimiento; ayudando con esto a impulsar el crecimiento económico del país, al mismo tiempo que la sociedad resulta beneficiada. Además, un análisis del London Econimics acerca de las actividades industriales en el país, confirmó que las imágenes de satélite son un mercado prioritario que apoya el crecimiento de una serie de industrias que hacen uso de estas; facturando un total de 234.7 billones de libas esterlinas por año, según datos del 2014. (London Economics, 2018)

4.7.2 Caso India.

En la India, uno de los países más poblados del mundo, han hecho diversos experimentos para explotar la información que extraída de imágenes de satélite; una de las pruebas más reciente fue la de diseñar un modelo que permitiera apoyar con la estimación del censo de población, antes de poner en práctica el censo oficial, de manera que permite a las oficinas de estadística de India planificar mejores estrategias para la realización del censo poblacional. (Nolte, 2010)

Las imágenes de satélite juegan un papel muy importante en las pruebas para el desarrollo del modelo para las mediciones intercensales de la India, debido a que como fuentes de información pueden cubrir grandes áreas de terreno que se pueden analizar; de manera que una vez procesadas exista la posibilidad de poder identificar y clasificar diferentes aspectos (clases) que faciliten la estimación de habitantes, por ejemplo, la extensión de ciudades. Además, una ventaja adicional que se tiene al trabajar con imágenes de este tipo es que, dependiendo de la tecnología del sensor, se pueden tener observaciones de un área específica cada determinado tiempo, por lo que mantener un monitoreo constante sobre el área elegida puede ser un trabajo sencillo. (Nolte, 2010)

El desarrollo de este tipo de modelos permite que la generación de datos de población contenga diferentes resoluciones espaciales y diferentes detalles de información adicionales en cada capa de la imagen, además de que la cantidad de datos de entrada aumenta el detalle de la información resultante; un ejemplo de los datos de entrada pueden ser los resultados de alguna de las encuestas anteriores realizadas en el país. (Nolte, 2010)

4.7.3 Caso México.

En México podemos ver otro caso de uso para las imágenes de satélite, el INEGI inició un proyecto que tiene como objetivo la identificación de cultivos a partir de imágenes satelitales, con el fin de determinar si es viable la generación de estadísticas con este método para el país. (Medina, 2018)

El proyecto inició en el año 2009 con el apoyo de NASS (National Agricultural Statistics Service) del departamento de agricultura de los Estados Unidos, StatCan (Statistics Canada) de Canadá y el JRC (Joint Research Centre) de la Unión Europea; estás instituciones ya utilizaban imágenes de satélite para generar información de estadísticas agropecuarias, por lo que apoyaron a INEGI con la implementación de este proyecto. (Medina, 2018)

Como resultado del proyecto, se estableció la metodología para la estimación de cultivos a partir de imágenes de satélite; dicha metodología se ha aplicado para la estimación de superficies con sorgo, limón, uva, papa, arroz, trigo, maíz, frijol, avena, nopal, naranja y plátano; alcanzando hasta un 95% de exactitud en sus pruebas, comparando lo que su algoritmo determinó con los datos registrados en SAGARPA (Secretaria de Agricultura y Desarrollo Rural), organismo encargado de recolectar la información agrícola en el país. Sin embargo, se determinó que existen casos en los que la metodología no es aplicable, por ejemplo, cuando se intenta estimar las superficies que contienen una combinación de diferentes cultivos. (Ponce Medina, 2018)

La metodología requiere que se cuente con buenas imágenes de satélite, es decir, que tengan poco o nada de nubosidad, ya que de lo contrario se presentaran problemas con el análisis y el resultado de la evaluación. Este problema se presenta mayormente al momento de querer apreciar cultivos que se dan en zonas con mucha lluvia durante el año, por lo que la estimación de estos cultivos no es viable. (Ponce Medina, 2018)

4.8 Cubo de datos.

El termino cubo de datos se utilizó originalmente en el proceso analítico de datos comerciales y estadísticos; el cubo de datos representa una matriz multidimensional de metadatos, que, en conjunto describen la semántica de ejes, coordenadas y celdas. Recientemente ha surgido el termino en un contexto geoespacial; con un enfoque hacia la gestión y análisis de grandes volúmenes de información que tienen un crecimiento rápido. (Geoscience Australia, 2019)

En la década de 1980, se buscó implementar el primer cubo de datos geoespacial, sin embargo, solo se utilizaron imágenes con datos hiperespectrales debido a que no se contaba con la tecnología para almacenar, procesar y analizar eficientemente la información. (Geoscience Australia, 2019)

Un cubo de datos geoespaciales se basa en datos espaciales y/o temporales que se dividen en regiones; estas regiones pueden ser cuadriculas regulares o irregulares, que permiten ubicar zonas para procesar y llevar a cabo un análisis específico dentro de la misma. El objetivo de un cubo de datos geoespacial es permitir la ingestión, el almacenamiento, el suministro y análisis de datos geoespaciales estructurados; por lo cual un cubo de datos debe de cubrir varios aspectos técnicos, conocidos como “las caras del cubo”: (Baumann, Lewis, & Szantoi, 2017)

1. Modelo de Parámetro.

Un valor de celda en el cubo de datos se describe mediante un modelo de parámetro que permite comprender la información almacenada en cada capa del cubo;

esto incluye la parametrización de la propiedad y su calidad, así como los metadatos necesarios para los análisis asociados a cada imagen. El Consorcio Geoespacial Abierto (OGC), mediante el Modelo de Datos Comunes (CDM), define elementos importantes de los modelos de parámetros, por ejemplo, la elevación del terreno es una de las implementaciones mejor documentadas de dichos modelos.

El hecho de incorporar datos que describan los mismos datos de parámetros, pero de diversos orígenes en un cubo de datos geoespaciales sigue siendo un desafío, debido a las diferencias entre los sensores de recolección, las cadenas de procesamiento de imágenes y los algoritmos utilizados, por lo que, dichos datos geoespaciales deben procesarse previamente con algoritmos aprobados por el Comité de Satélites de Observación de la Tierra (CEOS).

2. Representación de datos.

La representación de datos es la forma en que un parámetro se discretiza y codifica semánticamente a lo largo de los diferentes ejes o dimensiones del cubo, como el espacio, el tiempo y las propiedades de análisis. Un parámetro dado puede representarse de diferentes maneras y el mismo esquema de representación puede usarse para diferentes parámetros. Dependiendo del tipo de representación, se debe proporcionar un conjunto específico de metadatos que incluyen, por ejemplo, rango, intervalo, escala, precisión o referencia.

La Organización Internacional de Normalización (ISO) y el OGC basan la mayoría de sus cuadrículas, para el análisis de regiones, en el catálogo de proyecciones EPSG.

3. Organización de datos.

Los valores de celda generados por la discretización del parámetro deben estar dispuestos físicamente y almacenados de forma legible por máquina; esto implica a los

formatos de archivo, sistemas de archivos y estructuras de bases de datos. El OGC adopta la norma ISO 19123-2 que establece cómo la representación de los datos puede basarse en ASCII, con formatos como: GML, JSON o RDF; en binario, como GeoTIFF o NetCDF; o en una mezcla de ambos, haciendo uso de algún "formato contenedor", como zip o GeoPackage.

4. Infraestructura.

Las unidades de almacenamiento de datos deben estar alojadas en una infraestructura de TI y tener una configuración centralizada o distribuida de dispositivos de almacenamiento y procesamiento. Un punto importante a considerar es el acceso y la transferencia eficiente de datos entre las instancias de almacenamiento y procesamiento.

La cantidad y el aumento de datos geoespaciales requieren importantes inversiones financieras y logísticas para ofrecer servicios competitivos para atraer y retener usuarios. Entre las muchas instalaciones de supercomputación, que en los últimos años han comenzado a ofrecer datos y servicios geoespaciales, se encuentran iniciativas comerciales como Google Earth Engine y Amazon Web Services. Otros son financiados y operados con fondos públicos, como el Cubo de Datos Geoespaciales de Australia, el Centro de Datos de Observación de la Tierra de la Universidad Técnica de Viena (EODC) o la Plataforma de Procesamiento de Datos de Observación de la Tierra del JRC (JEODPP) en la Comisión Europea (CE) .

En el marco del programa Copérnico, la CE está a punto de financiar varios consorcios que unen a entidades públicas y privadas para servir como "Sistemas de información y acceso a datos" (DIAS)]. Si bien todas estas iniciativas también muestran compromisos que cubren otros aspectos de los cubos de datos, sus principales inversiones parecen estar dirigidas a la infraestructura de TI. Sin embargo, el éxito de estas inversiones dependerá en gran medida de la funcionalidad de estas

infraestructuras para las cuales también deben cubrir debidamente las otras caras descritas aquí.

5. Acceso y Análisis.

Dentro de la infraestructura, se debe implementar una amplia gama de funcionalidades a través del software para acceder, manipular y analizar los datos almacenados, y con esto poder ingerir nuevos productos en el cubo de datos. Estas funcionalidades deben documentarse y ponerse a disposición de los usuarios, lo cual podría lograrse haciendo uso de una API y algunas interfaces interactivas.

Entre la API de usuario (front-end) y las rutinas de manipulación de archivos (back-end) se pueden crear una o varias capas de software. Una de estas capas podría consistir en herramientas GIS, tales como QGIS o ArcGIS, y los Servicios de cobertura web del OGC se pueden usar para conectarlos dentro del cubo de datos.

6. Interoperabilidad.

La interoperabilidad y la fusión escalable de información espacial a través de diferentes cubos de datos es crucial y altamente dependiente del uso de estándares internacionales sólidos que rigen los protocolos de acceso y transferencia para la comunicación entre el cliente y el servidor, así como entre diferentes servidores. La norma ISO 19123 define a un modelo de cubo de datos abstractos como parte del concepto de cobertura; sin embargo, debido a su nivel de abstracción aún no es interoperable. Su estándar hermano, OGC CIS 1.1 / ISO 19123-2, establece codificaciones concretas que permiten volver a convertir las coberturas de un formato a otro para que sea posible un intercambio de cubos de datos independiente del formato bien definido, aunque a costa de una interpolación adicional.

4.8.1 Caso del Cubo de Datos de Geoscience Australia.

El objetivo del cubo de datos, desarrollado en Geoscience Australia, es aprovechar todo el potencial de los datos provenientes de las observaciones de la Tierra desde el espacio. Este proyecto involucra varios desafíos con relación a Big Data, tales como: el volumen de almacenamiento, la velocidad de procesamiento y el análisis de resultados; de manera que no se limite la utilidad de los datos. (Lewis, y otros, 2017)

Después de varios meses de trabajo, el cubo de datos en su segunda versión demuestra un avance importante en el cumplimiento de sus objetivos; sus fundamentos y componentes principales son: (Lewis, y otros, 2017)

1. *Preparación de datos.* Esto incluye correcciones geométricas y radiométricas a imágenes de satélite, para producir mediciones de reflectancia de superficie estandarizadas que admiten análisis de series de tiempo y sistemas de gestión de recolección; mismas que rastrean la procedencia de la imagen y/o producto del cubo de datos. Además, también se ve involucrada en la formalización de las decisiones sobre el procesamiento.

2. *El entorno de software.* Este es el componente que se encarga de la gestión e interacción con los datos; en términos más técnicos, este componente es el encargado de administrar la base datos y, a partir de ella, generar los diferentes productos que se pueden obtener de un cubo de datos.

3. *La infraestructura de hardware.* Este componente es de suma importancia, ya que es la base para hacer posible el funcionamiento del software; el hardware que se utilice para la implementación del proyecto debe ser capaz de almacenar grandes volúmenes de datos, tenerlos en línea para su uso inmediato y el poder de procesamiento para generar productos lo antes posible.

El enfoque del cubo de datos está en permitir a los analistas, de diferentes ámbitos, poder extraer nueva información rica en series de tiempo, incluso aprovechando nuevos métodos que mejoran el procesamiento de las coberturas espaciales y temporales sobre los datos de observación de la Tierra. Un punto importante a favor del código del cubo de datos es que se basa en una licencia Apache 2.0, lo que permite a desarrolladores externos, o incluso grandes organizaciones, a explorar el código y mejorarlo para conseguir nuevos resultados y/o nuevos productos. (Lewis, y otros, 2017)

Actualmente el Cubo de Datos de Geoscience Australia ha demostrado ampliamente su capacidad para aprovechar los datos provenientes de observaciones de la Tierra, a través de un número creciente de aplicaciones que permite a diferentes tipos de usuarios operativos y de investigación, tener acceso a este tipo de datos, ya sea procesados o en crudo. De igual manera, la infraestructura sobra la que está trabajando el cubo de datos ha demostrado tener una arquitectura que puede permitir a los usuarios a facilitar el acceso, administrar y aprovechar al máximo el gran volumen de información sobre datos de observación de la Tierra que contiene. (Lewis, y otros, 2017)

4.8.2 Caso del Cubo de Datos en Suiza.

Debido a la necesidad de monitorear los recursos ambientales, un conjunto de universidades y centros de investigación en Suiza determinó optar por hacer uso de imágenes de satélite aprovechando que se encuentran cada vez más disponibles y con una mejor difusión, a través de repositorios de acceso libre y abierto. Sin embargo, el hacer uso de datos provenientes de observaciones de la Tierra es un riesgo importante debido a su complejidad, volumen creciente y la falta de capacidades de procesamiento eficiente. (Giuliani, y otros, 2017)

El proyecto inició cuando se informaron del avance que se había tenido en Australia en cuanto al uso de este tipo de herramienta; y es que, no es para menos que el hecho de hablar de la implementación de un cubo de datos geoespacial, capaz de soportar los desafíos de Big Data y poder proporcionar un acceso sencillo a grandes

volúmenes de datos espaciotemporales en un formato listo para el análisis (ARD). CEOS define ARD como: (Baumann, Lewis, & Szantoi, 2017)

"Datos satelitales que han sido procesados con un conjunto mínimo de requisitos y organizados en una forma que permite el análisis inmediato sin esfuerzo adicional del usuario"

Los cubos de datos geoespaciales en general se crearon con el objetivo de aprovechar todo el potencial de los repositorios de imágenes de satélite, enfrentado los desafíos de volumen, velocidad, procesamiento y acceso libre a los datos. Para abordar los desafíos impuestos por Big Data, es necesario cambiar la mentalidad en cuanto al uso del procesamiento tradicional y empezar a usar métodos de procesamiento distribuido. Actualmente existen varios cubos de datos en operación y cada uno de ellos ha enriquecido con experiencia como enfrentar los diferentes desafíos con los que se puede encontrar. Una sugerencia que se plasmó durante el desarrollo del cubo de datos suizo fue crear un conjunto de archivos por escena, con el fin de solucionar las complejidades relacionadas con la preparación, el manejo, el almacenamiento y análisis de los datos; esta sugerencia si fue implementada con éxito y fue una variante más de la implementación de un cubo de datos. (Giuliani, y otros, 2017)

Al mismo tiempo que se trabajaba en el desarrollo del cubo de datos suizo, otro reto que se enfrentaron los investigadores fue el hecho de pre-procesar las imágenes y es que, hasta antes de diciembre del año 2018, la provisión sistemática y regular de imágenes de satélite ARD era un tema complejo, debido a que ninguna agencia espacial o distribuidora de imágenes de este tipo los ofrecía. Esto ocasionaba que el hecho de querer contar con una cobertura uniforme y consistente fuera una tarea difícil, por lo que se dieron a la tarea de desarrollar herramientas que facilitaran dicho proceso. Sin embargo, a partir de diciembre 2018 las agencias espaciales empezaron a distribuir las imágenes ARD, lo que facilitó el proceso de recolección de datos de manera eficiente; así como también el procesamiento de estos. (Giuliani, y otros, 2017)

4.8.3 Caso del cubo de datos en Colombia.

En Colombia decidieron unirse a la comunidad del cubo de datos, por lo que emprendieron la construcción de uno, al que llamaron CDCOL; la construcción de CDCOL se creó con el objetivo de que, aprovechando lo avances que ya se tenían por parte de Geoscience Australia, los analistas expertos de imágenes de satélite puedan visualizarlas en 4 dimensiones: latitud, longitud, tiempo y espectral. Hacer uso de un cubo de datos permite tratar a las imágenes de satélite como una matriz multidimensional que permite al usuario, entre otras cosas, almacenar, consultar y procesar las imágenes de satélite. (Ariza-Porras, y otros, 2017)

Un caso de aplicación que se muestra por parte de los colombianos es que los analistas ambientales trabajan a menudo con un conjunto de imágenes satelitales, que ellos mismos seleccionan, descargan y procesan. La idea del cubo de datos, acerca de poner todo “en un solo lugar”, permite que los analistas mejoren su productividad y tiempo en la generación de resultados; mismos que pueden ser reutilizados para la generación de otro producto. (Ariza-Porras, y otros, 2017)

En Colombia, existe el Sistema de Monitoreo de Bosques y Carbono (SMBYC), el cual es encargado de apoyar en la formulación, implementación y evaluación de políticas ambientales. Además, proporciona a entidades de gobierno reportes relacionados con el monitoreo de la cobertura forestal, la deforestación, las reservas de carbono forestal y las emisiones de carbono de la deforestación. Para lograr la correcta generación de estos reportes, el SMBYC requiere de procesar las imágenes de satélite disponibles sobre todo el país para un año específico y obtener resultados oportunos, consistentes y rentables. (Ariza-Porras, y otros, 2017)

Como se ha visto en los casos anteriores, el procesamiento de imágenes satelitales sobre un cubo de datos requiere de una infraestructura con una gran capacidad de almacenamiento y procesamiento, además de, contar con el personal adecuado para su correcto funcionamiento. (Ariza-Porras, y otros, 2017)

5. DISEÑO DE LA INTERVENCIÓN

La metodología para el análisis y diseño de sistemas propuesto por A. Cáceres; explica que existe una serie de etapas a seguir para la construcción de un sistema de información, las cuales se deben completar en la mayoría de lo posible para asegurar un correcto funcionamiento del sistema; estas etapas se encuentran divididas en 4 fases que permiten un mejor entendimiento y organización de las mismas. La siguiente lista que se muestra a continuación, desglosa las 4 fases con sus respectivas actividades o etapas: (Caceres, 2014)

1. Fase de Inicio:
    a. Investigación Preliminar.
    b. Análisis de los problemas a resolver.

2. Fase de Diseño:
    a. Escenarios de uso.
    b. Diseño del sistema.

3. Fase de Construcción:
    a. Construcción e implementación del sistema.

4. Fase de pruebas:
    a. Evaluación del sistema

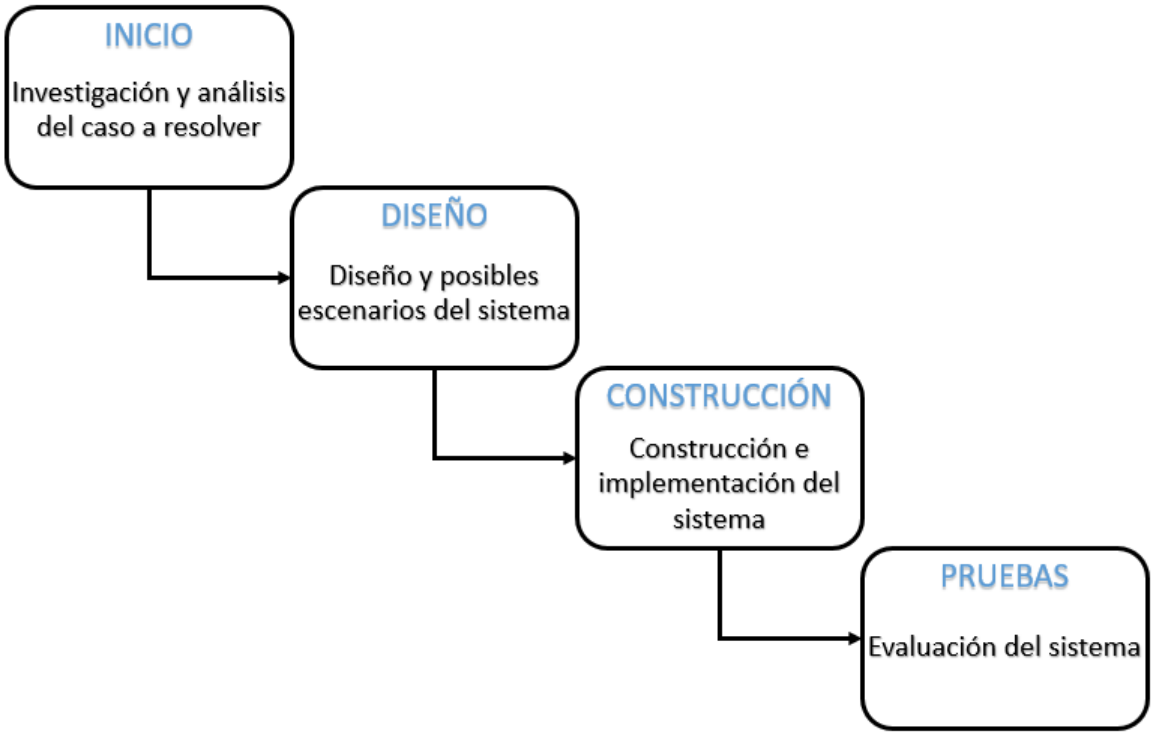


Figura 6. Diagrama de las fases para el diseño de la intervención.

Con base en la metodología mencionada, se muestra a continuación el desarrollo de las etapas para la resolución del caso práctico que presenta este documento.

5.1 Fase de inicio.

Como se mencionó en la sección 2 del presente documento, las necesidades o problemas a resolver, para este caso práctico, implicaron el poder medir los avances de los objetivos de desarrollo sostenible a partir de una nueva fuente información, las imágenes de satélite.

Con el fin de facilitar y mejorar en varios aspectos la medición de los ODS, el organismo encargado de llevar a cabo esta tarea, el INEGI, inició este proyecto que plantea la posibilidad de hacer uso de las imágenes de satélite para poder obtener

distintos tipos de datos, tales como: El crecimiento de área urbanas, los cuerpos de agua en el país, el índice de vegetación, la detección de cultivos, entre otros más.

Pensar en trabajar con este tipo de fuente de información, implicó tener en cuenta el espacio de almacenamiento que se requiere para contener un gran acervo de imágenes de satélite, dado que fue necesario contar con imágenes históricas y actuales del país, que van desde el año de 1984 hasta la actualidad (2019); contar con un acervo de esta magnitud permite no solo generar información actual, sino que a su vez se tiene la posibilidad de medir los cambios que ha sufrido el territorio nacional a través del tiempo.

Otro aspecto que debió ser considerado al momento de trabajar con imágenes de satélite, como ya ha sido mencionado anteriormente, es el poder de procesamiento; dado que la generación de productos a partir de este tipo de imágenes, requieren de un gran poder de equipo de cómputo que solo podría ser brindado por un servidor o estaciones de trabajo, ya que estamos hablando de máquinas que cuentan con más de un procesador y una cantidad suficiente de memoria RAM, para cumplir con los objetivos.

La generación de los productos con la información requerida para apoyar en la medición de los ODS o a alguna otra área del Instituto que la requiera; necesita del software capaz de hacer el trabajo. Debe ser un software que encuentre el equilibrio entre el uso de procesadores y memoria, quizá haciendo uso de un lenguaje de programación sencillo y potente; con el fin de que pueda ser mejorado por algún otro equipo de programadores, y que a su vez permita la manipulación por usuarios o algún experto del área. Este fue el último aspecto considerado, en general, para el diseño de la arquitectura de este proyecto.

5.2 Fase de Diseño.

En esta fase de la intervención se planteó realizar una búsqueda de alternativas que apoyen con la resolución de las necesidades que se presentaron en el punto anterior.

Se llevó a cabo un análisis de cada una de las posibles soluciones y se seleccionó la mejor opción, es decir, aquella que tomó en cuenta los aspectos de hardware y software que cumplan con los objetivos de este proyecto.

5.2.1 Elección de software.

El primer aspecto analizado, antes de empezar con el diseño de la arquitectura de hardware para el proyecto, fue decidir cuál sería el software indicado para trabajar de manera rápida y eficaz con imágenes de satélite, de esta manera se podría asegurar que el hardware que se definiera sería el apropiado para el mejor funcionamiento del software y con ello optimizar la generación de resultados.

Después de una búsqueda por la web de posibles herramientas de software que permitieran el trabajo con imágenes de satélite; se decidió analizar a profundidad algunas que resaltaron por su reputación, experiencia e historia; se tomaron en cuenta herramientas con licenciamiento, herramientas de código abierto e incluso se consideró la posibilidad de construir una propia a partir de una con licencia Apache 2.0. En las siguientes líneas de este documento se exponen las alternativas encontradas, así como una breve descripción de estas:

5.2.1.1Herramientas GIS (ArcGIS, QGIS)

Un GIS (Sistema de Información Geográfica; por sus siglas en inglés) es una integración organizada de hardware, software y datos de información geográfica; dicha organización está diseñada para capturar, almacenar, manipular, analizar y desplegar en todas sus formas a la información geográficamente referenciada, con el fin de resolver problemas complejos. (Benz, Hofmann, Willhauck, Lingenfelder, & Heynen, 2004)

El GIS funciona como una base de datos con información geográfica que se encuentra asociada por un identificador común a los objetos gráficos de un mapa digital; de esta forma es posible conocer los atributos de localización de un objeto en la cartografía. (Benz, Hofmann, Willhauck, Lingenfelder, & Heynen, 2004)

El sistema GIS permite separar la información en diferentes capas temáticas y las puede almacenar independientes una de otras, lo que facilita el trabajar con ellas de manera más rápida y sencilla, ya que a su vez un experto en el sistema tiene la posibilidad de relacionar información existente a través de la topología de los objetos y con ello poder generar nuevas capas de información. (Benz, Hofmann, Willhauck, Lingenfelder, & Heynen, 2004)

Las principales cuestiones que puede resolver un Sistema de Información Geográfica, ordenadas de menor a mayor complejidad, son: (Benz, Hofmann, Willhauck, Lingenfelder, & Heynen, 2004)

- Localización: Preguntar por las características de un lugar específico.
- Condición: El cumplimiento o no de condiciones impuestas al sistema.
- Tendencia: Comparación entre situaciones temporales o espaciales distintas de alguna característica.
- Rutas: Calculo de rutas optimas entre 2 o más puntos.
- Pautas: Detección de pautas espaciales.
- Modelos: Generación de modelos a partir de fenómenos o actuaciones simuladas.

Existen varios métodos utilizados para la creación u obtención de datos digitales; el más utilizado es el de la digitalización, donde a partir de un mapa impreso que se transfiere a un medio digital es posible añadirle información georreferenciada. Además, dada la amplia disponibilidad de imágenes orto-rectificadas, tanto de satélite como aéreas, la digitalización por esta vía se está convirtiendo en la principal fuente de extracción de datos geográficos; esta forma de digitalización implica la búsqueda de datos geográficos directamente en las imágenes, en lugar de obtenerlos a partir del método tradicional de localización de formas geográficas sobre un tablero de digitalización. (Benz, Hofmann, Willhauck, Lingenfelder, & Heynen, 2004)

Los datos GIS representan los objetos del mundo real, como carreteras, suelo, altitudes, entre otras cosas. Los objetos del mundo real se pueden dividir en 2 abstracciones: Los objetos discretos, como una casa o un edificio; y los objetos continuos, como la cantidad de lluvia que cae o la elevación de un cerro. En los Sistemas de Información Geográfica es posible almacenar ambas abstracciones, y se les llama imágenes ráster e imágenes vectoriales. (Benz, Hofmann, Willhauck, Lingenfelder, & Heynen, 2004)

Los GIS que se enfocan en el manejo de datos en forma vectorial son los más populares en el mercado, sin embargo, aquellos que son capaces de procesar imágenes ráster son muy utilizados en estudios que requieran la generación de capas continuas, como en estudios medioambientales donde no se requiera una excesiva precisión espacial; por ejemplo, la contaminación atmosférica, la distribución de temperaturas, la localización de especies marinas, análisis geológicos, entre otras tantas más. (Benz, Hofmann, Willhauck, Lingenfelder, & Heynen, 2004)

5.2.1.2 Open Data Cube (ODC)

El ODC es un software de código abierto que permite acceder, administrar y analizar grandes cantidades de datos de observaciones de la Tierra. Una de sus grandes ventajas, es que fue desarrollado con el fin de analizar series de tiempo sobre estas observaciones, sin embargo, la flexibilidad de la plataforma también permite que se incluyan y analicen otro tipo de colecciones de datos; dichos datos pueden incluir modelos de elevación, rejillas geofísicas, superficies interpoladas y salidas de modelos. (Geoscience Australia, 2019)

Una característica clave del ODC es que mantiene cada capa como una observación única, lo que contrasta con muchos otros métodos utilizados para manejar grandes colecciones de datos georreferenciados. Algunas de las principales ventajas que tiene este software son las siguientes: (Geoscience Australia, 2019)

- Marco flexible.
- El usuario mantiene el control y la propiedad sobre sus datos.
- Cambio de paradigma de análisis basado en escena a uno basado en pixeles.
- Barrera de entrada inferior para el análisis de datos de teledetección.

El proyecto del cubo de datos surge de la necesidad de gestionar mejor los datos satelitales; el software es capaz proporcionar una base de varias soluciones de arquitectura de datos a escala internacional, regional o nacional. El cubo de datos funciona bien con Analysis Ready Data (ARD); datos pre procesados y listos para el análisis, dichos datos se encuentran puestos a disposición en la nube por distintos proveedores que trabajan por tener los productos ARD globales. El sistema del ODC, está diseñado principalmente para: (Lewis, y otros, 2017)

- Catalogar grandes colecciones de datos de observación de la Tierra.
- Proporcionar una API basada en Python para consultas y acceso a los datos que demanden un alto poder de cómputo.
- Brindar a los científicos y a otros usuarios la capacidad de realizar fácilmente un análisis de datos exploratorio.
- Permitir el procesamiento escalable de los datos almacenados.
- Rastrear a procedencia de todos los datos contenidos para permitir el control de calidad y de las actualizaciones.

El núcleo del ODC sirve como una capa entre los proveedores de datos satelitales y las aplicaciones de código abierto que existen como herramientas para ayudar a los científicos a realizar investigaciones utilizando datos administrados por el ODC; a continuación, se listan las herramientas más populares utilizadas dentro de la comunidad que utiliza el ODC como base: (Lewis, y otros, 2017)

- *Herramientas de líneas de comando:* Una herramienta utilizada por programadores/desarrolladores para interactuar con el ODC.
- *Open Data Cube Explorer:* Una aplicación web visual e interactiva que permite a los usuarios explorar su inventario de datos disponible.
- *Open Data Cube Stats:* Es un medio optimizado para definir y ejecutar análisis avanzados en el sistema del ODC. Esta herramienta está orientada a los científicos.
- Interfaz de usuario web: Una aplicación web que permite a los desarrolladores mostrar y visualizar interactivamente la salida de algoritmos.
- Jupyter Notebooks: Documentos de investigación centrados en técnicas de ciencias en observaciones de la Tierra. Un cuaderno contiene código ejecutable que detalla ejemplos de cómo se usa el cubo de datos en un entorno de investigación.
- Servicios web de Open Geospatial Consortium (OGC): Adaptadores que pueden conectar aplicaciones que no son ODC al ODC.

5.2.1.3 Toma de la decisión

Una vez concluida la búsqueda e investigación de posibles herramientas que pudiesen apoyar con la resolución de las necesidades planteadas anteriormente, se tomó la decisión de elegir el software del cubo de datos (Open Data Cube, ODC), ya que nos ofrecía mayores ventajas que las herramientas GIS. Algunas de las ventajas que inclinaron la balanza a favor del ODC se listan a continuación:

- El software del ODC es de código abierto, por lo que es posible adaptarlo a necesidades particulares, así como extender sus capacidades.
- Permite trabajar con grandes colecciones de imágenes accediendo a ellas por medio de consultas a una base de datos.
- Su filosofía sobre el análisis por píxel permite tener un análisis más profundo sobre una escena, incluso aunque en esta predomine la nubosidad de la zona de interés.

- El software de complemento con el que cuenta el cubo de datos es bastante atractivo para hacer experimentos y descubrir nuevos temas de investigación.

Una vez determinado el software que apoyaría en la realización del caso práctico, se prosiguió con el diseño de la arquitectura de hardware para que el cubo de datos trabajara de la forma más eficiente y óptima posible.

5.2.2 Diseño de Arquitectura para el hardware.

Para que un software alcance sus niveles óptimos de funcionamiento debe de estar instalado sobre una infraestructura de hardware que se acoplé de la mejor manera a los requerimientos que este solicite; para el caso particular del cubo de datos no se requiere de grandes máquinas para ser instalado, ya que su escalabilidad le permite estar tanto en equipos de cómputo personales como en supercomputadoras o grandes servidores; sin embargo, para conseguir un desempeño apropiado por parte del cubo de datos es necesario contar con una infraestructura de hardware ajustada especialmente para el manejo de imágenes de satélite.

Para el desarrollo de este caso práctico se realizaron un par de diseños, a manera de propuestas, con el fin de poder seleccionar aquel que podría ser la mejor opción, considerando todo aquello que se ha venido mencionando desde el inicio de este documento; teniendo a la capacidad de almacenamiento y al poder de procesamiento, como las características más importantes.

Como consideraciones iniciales para las propuestas de los diseños de arquitecturas, se consideraron los siguientes datos sobre el conjunto inicial de imágenes de satélite a procesar que el INEGI recibió en Marzo de 2019:

- El acervo total de imágenes de satélite Landsat, que cubren toda la superficie continental de México, contiene 109,668 imágenes.
- Las imágenes abarcan un periodo de tiempo del año de 1984 hasta el 2018.

- Fueron captadas por 4 diferentes sensores a lo largo de la historia; Landsat 4 (ls4), Landsat 5 (ls5), Landsat 7 (ls7) y Landsat 8 (ls8).
- El espacio de almacenamiento total que ocupa este acervo de imágenes comprimidas es de 29.2 TB.
- Las imágenes sin comprimir ocupan un espacio de almacenamiento aproximado de 90 TB.

Las propuestas que se generaron fueron:

1. Servidor virtualizado con VMware.

Esta propuesta nace de una idea de intentar mantener el proyecto dentro de algo simple y tradicional; un único servidor que se encuentre conectado a un almacenamiento de tipo NAS para contener las imágenes. Las características que se plantearon para este servidor son las siguientes:

- Sistema Operativo: Linux Ubuntu 16.04.
- 32 núcleos de procesamiento.
- Memoria RAM: 64 GB.
- Memoria Interna: 100 GB.
- Almacenamiento NAS: 100 TB.

La siguiente figura muestra el diagrama de la propuesta:

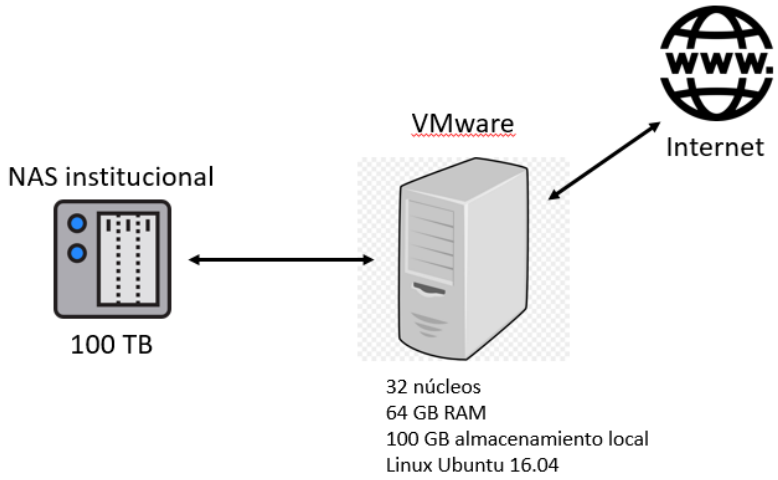


Figura 7. Diagrama de la arquitectura para el servidor de VMware.

2. Oracle Cloud Machine (OCM).

Esta se trata de una propuesta más elaborada que la anterior; pero en pocas palabras consiste en poseer una nube privada que incluya poder de procesamiento. Oracle ofrece el servicio de brindar el equipo necesario para soportar una nube privada, vendiendo por bloques el número de núcleos, la cantidad de memoria RAM y el espacio de almacenamiento, el cliente es quien debe considerar el número de bloques que se deben adquirir según el proyecto.

Para el caso específico de este proyecto, se analizó la posibilidad de adquirir una OCM con un bloque de especificaciones, es decir, que la nube privada de Oracle podría contar con lo siguiente:

- 108 TB de almacenamiento, en un arreglo de discos con RAID 6.
- 48 núcleos de procesamiento.
- 256 GB de RAM.
- Sistema operativo Oracle Linux 7.0
- Con posibilidad de conectar hasta 18 TB adicional de almacenamiento interno.

En la Figura 8 se muestra el diagrama de lo que sería la arquitectura de la OCM.

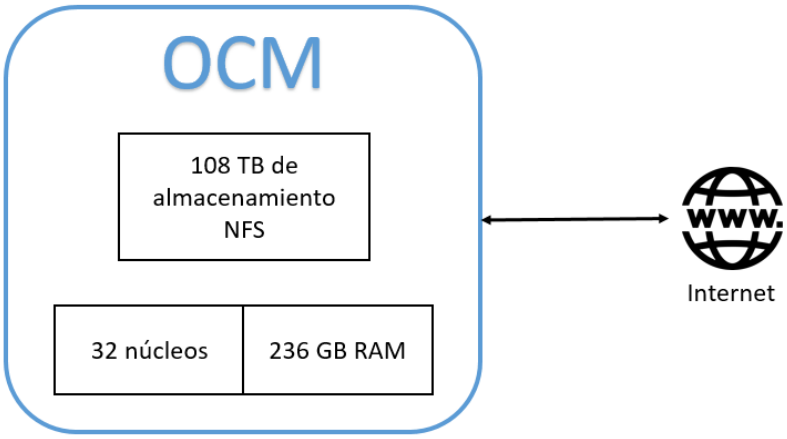


Figura 8. Diagrama de la arquitectura para el servidor de OCM.

5.2.3 Elección de diseño.

En cuanto a características del hardware presentado en los casos anteriores, las especificaciones de lo que nos puede ofrecer Oracle con una OCM lucían mucho mejor de lo que podría ser una arquitectura convencional para un proyecto, pero existía un punto a destacar entre estos 2 modelos de arquitecturas posibles para la realización del proyecto: El tema del costo; en el caso de un único servidor para llevar a cabo todo el trabajo, resultaba ser la opción más económica, dado que el Instituto no tendría que hacer un gran gasto en adquirir las tecnologías necesarias, pues ya contaba con varios similares. Sin embargo, hablar de una nube privada, con el poder de almacenamiento y de procesamiento que ofrece Oracle con su OCM, implica varios gastos adicionales que deberían ser ponderados presupuestalmente, no solo para este proyecto, sino para otros en el contexto que satisfacen las necesidades de una manera balanceada.

Después de unos días de análisis y pláticas entre directivos de INEGI, se aprobó la compra de los servicios de Oracle para poseer una nube privada dentro del instituto, de la cual se asignaron recursos para la realización de este proyecto y cuyo diseño “final” se muestra en la Figura 8.

5.3 Fase de Construcción e Implementación.

A principios de diciembre de 2018, un equipo especializado del fabricante (Oracle) se presentó en las instalaciones de INEGI con la encomienda de desplegar e integrar el OCM a la infraestructura del instituto.

El 19 de diciembre de 2018 se inició con el trabajo principal del caso práctico; la instalación e implementación de un cubo de datos geoespacial, el cual está planificado para contener la historia en imágenes de satélite que han sido capturadas sobre México entre los años de 1984 y 2019, siendo estas imágenes una donación al INEGI por parte de la USGSS (United States Geological Survey; Servicio Geológico de los Estados Unidos por sus siglas en inglés).

Los autores originales del cubo de datos son un grupo de científicos de Geoscience Australia; organismo gubernamental de aquel país encargado de apoyar al gobierno en temas de geografía y geología. Al ser ellos los creadores de las herramientas del cubo de datos, se estableció comunicación para platicar la idea y el hecho de querer hacer uso de su tecnología en el caso particular de nuestro país (México), acordando con ellos el apoyo en cuanto a la resolución de dudas y problemas que pudieran surgir durante el desarrollo de este proyecto; así mismo, compartieron la documentación del cubo de datos que se encuentra publicada en una plataforma digital sobre internet, dentro de la cual es posible encontrar una guía de instalación y una guía de uso rápido para el cubo de datos.

El fin de este documento no es mostrar los pasos a seguir para una correcta instalación del sistema, sin embargo, en el Anexo 1 se puede encontrar la bitácora de este proyecto en la cual se detallan los pasos que se siguieron para la instalación del software y la configuración de la base de datos. De igual manera, en la misma bitácora se detallan las acciones realizadas para conseguir el procesamiento de las imágenes y obtener productos a partir de ellas; la descripción de los productos procesados, así como los detalles de los tiempos de procesamiento, se muestran en las siguientes secciones de este documento.

5.3.1 Productos procesados. Descripción.

El software del cubo de datos tiene varias herramientas adicionales con las que es posible obtener distintos productos, según sea el objetivo de cada proyecto/investigación. Una de las herramientas adicionales lleva como nombre datacube-stats (Datacube Statistic) y será la que emplearemos con el fin de alcanzar los objetivos de este caso práctico. La siguiente tabla nos muestra algunos de los productos que es posible procesar a partir de dicha herramienta y las bandas espectrales que requieren de las imágenes para su procesamiento.

Tabla 1: Descripción de productos del cubo de datos

Productos	Descripción
Geomediana (GM)	Mediana geométrica por pixel. Cuenta con 6 bandas espectrales básicas: • Blue (Azúl) • Red (Rojo) • Green (Verde) • NIR (Infrarrojo) • SWIR1 (Infrarrojo de onda corta 1) • SWIR2 (Infrarrojo de onda corta 2)
NDVI **Normalized Difference Vegetation Index**	Índice diferencial normalizado de vegetación por pixel. La fórmula para su cálculo es la siguiente: NDVI= (NIR - Red) / (NIR + Red) Las bandas espectrales de resultado que se obtienen son: • ndvi_max • ndvi_mean • ndvi_median • ndvi_min • ndvi_std

NDBI **Normalized Difference Built-up Index**	Índice diferencial normalizado de construcción por pixel. La fórmula para su cálculo es: NDBI= (SWIR1 – NIR) / (SWIR1 + NIR) Las bandas espectrales de resultado son las siguientes: • ndbi_max • ndbi_mean • ndbi_median • ndbi_min • ndbi_std
MNDWI **Modified Normalized Difference Water Index**	Índice diferencial normalizado de agua modificado. Su fórmula para calcular este producto es: MNDWI= (Green – SWIR1) / (Green + SWIR1) Las bandas espectrales de resultado son las siguientes: • mndwi_max • mndwi_mean • mndwi_median • mndwi_min • mndwi_std
TCWBG **Tasseled Cap Transformation**	Optimización estandar para detectar humedad, brillo y verdor por pixel. Sus bandas de resultado son las siguientes: • pct_exceedance_brightness Porcentaje excedente de brillo. • pct_exceedance_greenness Porcentaje excedente de verdor • pct_exceedance_wetness Porcentaje excedente de humedad • mean_brightness Media de brillo • mean_greenness

	Media de verdor • mean_wetness Media de humedad • std_brightness Desviación estándar de brillo • std_greenness Desviación estándar de verdor • std_wetness Desviación estándar de humedad
UI **Urban Index**	Indice urbano por pixel. Se calcula mediante la siguiente fórmula: UI= (SWIR2 – NIR) / (SWIR2 + NIR) Tiene como resultado las siguientes bandas espectrales: • ui_max • ui_mean • ui_median • ui_min • ui_std
WOFS **Water Observation From Space**	Porcentaje de agua superficial por pixel. Sus bandas en el resultado son: • wet • wofs • total

5.3.2 Productos Procesados. Experimentación.

Una vez que se terminó con la instalación del cubo de datos sobre el servidor del OCM, fue necesario preparar la base de datos indexando todas las imágenes del acervo que fue donado por la USGS; para información más detallada del proceso de indexación a la base de datos se puede revisar el Anexo 1 de este documento, la bitácora que fue mencionada anteriormente.

Con el software en funcionamiento y la base de datos lista para usarse, se inició con la experimentación para la generación de productos sobre coberturas nacionales por año, desde 2000 a 2018. Para facilitar la generación de estos productos, se desarrolló un conjunto de scripts en Python que apoyaron con la automatización de este proceso; dado que, en la documentación describen el proceso de una manera poco óptima para generar productos de nivel nacional, pues una cobertura del país cuenta con 131 imágenes aproximadamente y en un año se alcanzan a recolectar poco más de 50 coberturas; alrededor de 6500 imágenes de satélite.

En el Anexo 2 de este documento se puede encontrar una descripción, con mayor profundidad de detalle, sobre como ejecutar el conjunto de scripts en Python que se desarrollaron para la automatización de la generación de productos. Además de que se explican las fases del proceso en general.

Se estableció que uno de los productos que más beneficios aportaría al INEGI sería la Geomediana, por lo que para las pruebas experimentales se decidió generar este producto para años claves; 2010, 2015 y 2018. La siguiente imagen muestra una tabla con los tiempos tomados por fase para generar la geomediana del año 2018, en lo que sería el primer intento para generar productos con el cubo de datos de México:

Fase	Transferir Archivos	Descompactar imágenes	Indexar imágenes	Ingestar imágenes	Generar productos	Total
1 (100 %)	Total: 6239 Tamaño: 2.1 Tb Velocidad: ~5.4 MB/s Tiempo: 106 H 46 Min	Inicio: 2.1 Tb Final: 6.2 Tb Tiempo: 118 H 19 Min				225:05
2 (100 %)			Tiempo: ~1H 50 Min	Inicio: 6.7 Tb Final: 914 Gb Tiempo: 327H 52 Min		329:42
3 *(6.2 %)*					Se canceló el proceso a un 6.19% transcurridos 23H 39 min	23:39
						578:26

Figura 9. Registro de tiempos utilizados para el primer intento de generar la Geomediana 2018.

Lamentablemente no se consiguió el objetivo, debido a los tiempos tan elevados que estaba tomando esta prueba; pues como se puede observar en la última fase se tenía un total de casi 24 horas y solo se había avanzado un 6%, lo que quiere decir que tomaría alrededor de 3 meses terminar la generación de la Geomediana.

5.3.3 Posible solución al experimento.

Era claro que se tenía un problema con el tiempo de procesamiento, se revisó incluso con los creadores del software del cubo de datos y se coincidió en que el problema era provocado en parte por la arquitectura de hardware. Haciendo un análisis de la arquitectura se llegó a la conclusión de que el problema radicaba en la velocidad de lectura y escritura del almacenamiento, pues el monitoreo de procesadores al momento de estar ejecutando la generación de productos no alcanzaba ni siquiera el 20%, en promedio, de su capacidad.

La solución que se propuso para atender esta problemática fue agregar un almacenamiento dedicado exclusivamente para colocar los datos a procesar en el momento, sin embargo, esta solución tenía una limitante, y es que el OCM solo permite agregar hasta 18 TB adicionales al almacenamiento dentro del bloque inicial que se contrató; este espacio adicional estaría constituido por discos de alta velocidad de lectura y escritura, modificando el diagrama de la arquitectura de hardware como se muestra en la Figura 10.

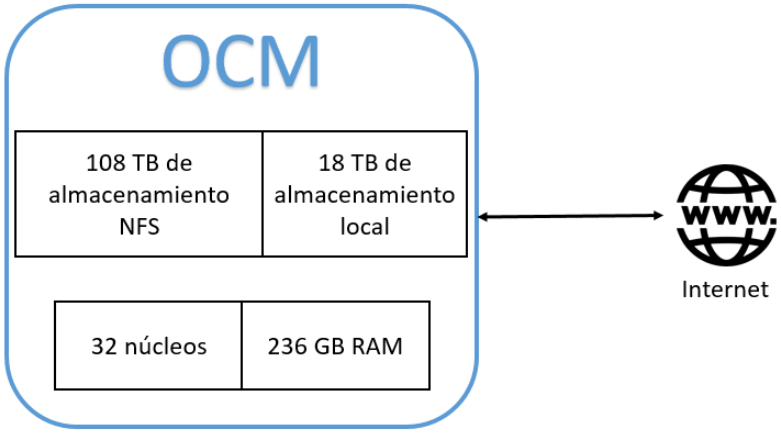


Figura 10. Diagrama de la arquitectura para el servidor de OCM mejorado.

6. EVALUACIÓN DE LA INTERVENCIÓN

6.1 Resultados Obtenidos de la experimentación.

Una vez que se implementó el almacenamiento exclusivo para los datos que se procesarían, se inició nuevamente el experimento para generar la Geomediana del año 2018.

Fase	Transferir Archivos	Descompactar imágenes	Indexar imágenes	Ingestar imágenes	Generar productos	Total
1 (100 %)	Total: 6239 Tamaño: 2.1 Tb Velocidad: ~13 Mb/s Tiempo: 63H 55 Min	Inicio: 2.1 Tb Final: 6.2 Tb Tiempo: 7H 09 Min				71:04
2 (100 %)			Tiempo: 01:03	NO se realizó ingestión		01:03
3 (100 %)					Tiempo: GM: 38 H 14 Min Resto: 28 H 11 Min	66:25
						138:32

Figura 11. Registro de tiempos utilizados para la generación de la Geomediana 2018.

En esta ocasión los resultados si fueron satisfactorios, pues permitieron terminar el proceso de la generación de la Geomediana para ese año en un tiempo relativamente corto si se compara con el intento anterior; ya que en casi 1 semana se podría contar con el producto terminado y no en los desalentadores 3 meses de la prueba inicial. Como se mencionó anteriormente, la intención de esta fase experimental también incluía hacer la Geomediana para los años 2010 y 2015, cuyos resultados se presentan en las siguientes páginas.

La Figura 12, muestra el resultado del procesamiento de la Geomediana para el año 2018; en palabras coloquiales, se obtuvo una imagen de satélite del territorio nacional sin nubes.

Figura 12. Geomediana 2018.

En la Figura 13, se muestra una tabla con los tiempos requeridos por procesamiento de la Geomediana para el año 2010. En ella se destaca que se ocupo 50% menos tiempo que la generación de la Geomediana 2018, sin embargo, esto se debe a la cantidad de imágenes tomadas entre un año y otro: 3707 imágenes para el año 2010 y 6239 imágenes tomadas para el año 2018.

Fase	Transferir Archivos	Descompactar imágenes	Indexar imágenes	Ingestar imágenes	Generar productos	Total
1 (100 %)	Total: 3707 Tamaño: 861 Gb Velocidad: ~10 Mb/s Tiempo: 27 H 12 Min	Inicio: 861 Gb Final: 3.6 Tb Tiempo: 9H 23 Min				36:35
2 (100 %)			Tiempo: 1 H 26 Min	NO se realizó ingestión		1:26
3 (100 %)					Tiempo: GM: 17 H 22 Min Resto: 14 H 15 Min	31:37
						69:38

Figura 13. Registro de tiempos utilizados para la generación de la Geomediana 2010.

La figura 14 muestra el resultado del procesamiento de la Geomediana del año 2010; para la cual se utilizaron imágenes capturadas con los satélites Landsat 5 y Landsat 7.

Figura 14. Geomediana 2010.

Fase	Transferir Archivos	Descompactar imágenes	Indexar imágenes	Ingestar imágenes	Generar productos	Total
1 (100 %)	Total: 6193 Tamaño: 2.1 Tb Velocidad: 16.6 Mb/s Tiempo: 48 H 41 Min	Inicio: 2.1 Tb Final: 6.2 Tb Tiempo: 7 H 44 Min Revisando md5: 3 H 37 Min Descompactando: 4 H 07 Min				56:18
2 (100 %)			Tiempo: 00:57	NO se realizó ingestión		00:57
3 (45 %)					Tiempo: Se ejecutaron de forma simultánea. GM: 48 H 38 Min Resto: 50 H 07 Min	50:07
						107:28

Figura 15. Registro de tiempos utilizados para la generación de la Geomediana 2015.

Las figuras 15 y 16, muestran los registros de tiempo y el resultado obtenido del proceso de generación de Geomedianas para el año 2015. En la figura 16, se resalta un error en uno de los cuadrantes de la imagen, pues se visualiza como un espacio nulo dentro del territorio nacional, es decir, falta información sobre las coordenadas específicas y puede deberse a la inexistencia de imágenes en la base de datos sobre ese punto.

Figura 16. Geomediana 2015.

Pensar en la generación de las Geomedianas de cada año, suena una tarea que requiere bastante tiempo, sin embargo, el beneficio que se puede obtener de ellas es mucho mayor. Un claro ejemplo de ello es el poder detectar y medir cambios en los ecosistemas, ciudades, cuerpos de agua, entre otros; y con estas mediciones apoyar a los gobiernos a tomar las mejores estrategias para el beneficio de la sociedad.

6.2 Propuesta de mejora para la arquitectura.

Con la mejora que se tuvo agregando el almacenamiento especial para los datos de procesamiento, se planteó la posibilidad de eliminar el almacenamiento interno de la OCM; un almacenamiento en red que estaba incluido en el bloque de recursos. La idea era no utilizar más ese almacenamiento y preparar un espacio sobre la SAN institucional de INEGI, donde se almacenaría el acervo de imágenes, sin embargo, es un tema que se debía que convenir con la gente de infraestructura del instituto para conocer las posibilidades de llevarlo a cabo.

Otro aspecto importante considerado con respecto al almacenamiento fue que los productos generados también ocupan espacio, por lo que era ideal considerar otro almacenamiento especial para estos archivos resultantes; el espacio que ocupan los 7 productos por año es de un poco más de medio terabyte.

El departamento encargado de la infraestructura del añadió por medio de una conexión NFS (Network File System; Sistema de Archivos por Red) al servidor del OCM, los almacenamientos propuestos. El siguiente diagrama muestra la arquitectura de hardware con las modificaciones realizadas; añadiendo los espacios de almacenamiento para el acervo de imágenes y los productos resultantes.

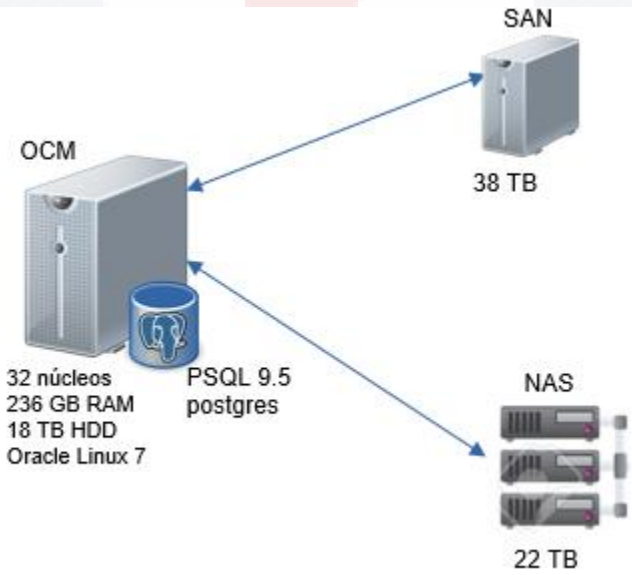


Figura 17. Nuevo diseño de la arquitectura de hardware.

6.3 Resultados de la propuesta de mejora aplicada.

Con las mejoras que se hicieron en la arquitectura se obtuvieron tiempos mucho más cortos que los que se mostraron en los 2 experimentos anteriores, se alcanzó una reducción del 80% del tiempo para la generación de las Geomedianas, logrando con esto tener un producto terminado en 24 horas aproximadamente. Este nuevo modelo de arquitectura permitió concluir con la generación de Geomedianas para los años del 2000 a 2019 en un mes.

Una vez que se conocieron los nuevos tiempos, se optó por realizar el proceso de ingestión de todas las imágenes que se tenían, se estimó un tiempo de espera de alrededor de 10 horas por año. Las imágenes transformadas en archivos netCDF permitieron la generación del resto de los productos en aproximadamente 20 horas cada uno, por lo que se pudo completar la generación de todos los productos del cubo de datos para cada año en un poco más de un mes.

BIBLIOGRAFÍA

Anderson, K., Ryan, B., Sonntag, W., Kavvada, A., & Friedl, L. (2017). Earth Observation in service of the 2030 Agenga for sustainable development. En *Geospatial Information Science* (págs. 77-96). Taylor & Francis Group.

Ariza-Porras, C., Bravo, G., Villamizar, M., Moreno, A., Castro, H., Galindo, G., . . . Lozano, P. (2017). *CDCol: A Geoscience Data Cube that Meets Colombian Needs.* Bogota: Springer International Publishing.

Baumann, P., Lewis, A., & Szantoi, Z. (2017). The six faces of the Data Cube. *Big Data from space*, (págs. 32-35). Toulouse, Francia.

Benz, U. C., Hofmann, P., Willhauck, G., Lingenfelder, I., & Heynen, M. (2004). Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information. *Photogrammetry & Remote Sensing*, 239 - 258.

Caceres, A. (2014). *Analisis y Diseño de Sistemas.*

Gartner, G., Huang, H., Research Group Cartography, & Vienna University of Technology. (2015). *Proceedings of the 1st ICA European Symposium on Cartography.* Vienna.

Giuliani, G., Chatenoux, B., De Bono, A., Rodila, D., Richard, J.-P., Allenbach, K., . . . Peduzzi, P. (2017). Building an Earth Observations Data Cube: lessons learned from the Swiss Data Cube (SDC) on generating Analysis Ready Data (ARD). En *Big Earth Data* (págs. 100 - 117). Taylor & Francis Group.

INEGI. (27 de 11 de 2018). *Institución con historia*. Obtenido de https://www.inegi.org.mx/inegi/quienes_somos.html

INEGI. (2019). *Memorias del proyecto Cubo de datos Geoespaciales.*

INEGI. (01 de 07 de 2019). *Principios Fundamentales de las Estadisiticas Oficiales*. Obtenido de https://www.inegi.org.mx/400.html?aspxerrorpath=/est/contenidos/proyectos/aspectosmetodologicos/principiosfundamentales/default.aspx

Lee, J.-G., & Kang, M. (2015). Geospatial Big Data: Challenges and Opportunities. *Big Data Research*, 74-81.

Lewis, A., Oliver, S., Lymburner, L., Evans, B., Wyborn, L., Mueller, N., . . . Wang, L.-W. (2017). The Australian Geoscience Data Cube — Foundations and lessons learned. *Remote Sensing Enviroment*, 276 - 292.

Lohr, S. (11 de 02 de 2012). The age of Big Data. *New York Times*, págs. 1-5.

London Economics. (2018). *Value of satellite-derived Earth Observation capabilities to the UK Government today and by 2020.* Londres.

Martinez, R., & Calvo, M. (01 de 07 de 2019). *Tipos de orbitas. Constelaciones satelitales*. Obtenido de Universidad Politécnica de Madrid: http://www.gr.ssr.upm.es/docencia/grado/csat/material/CSAT09-2-OrbitasConstelaciones.pdf

Medina, M. S. (14 de 02 de 2018). Proyecto de estimación de cultivos utilizando imágenes de satélite. (A. S. Valdez, Entrevistador)

Nolte, M. (2010). *The application of optical satellite imagery and census data for urban population estimation: A case study for Ahmedabad, India.* Karlsruher.

Ponce Medina, M. S. (2018). Estimación de cultivos usando imágenes de satélite., (págs. 1-5). Aguascalientes.

Richards, J. A., & Jia, X. (2006). *Remote Sensing Digital Image Analysis.* Berlin: Springer-Verlag.

UNECE. (2016). *In-depth review of developing geospatial information services based on oficial statistics.* Luxembourg.

United Nations, Australian Bureau of Statistics, Queensland University of Technology, Australia, Queensland Government, Australia, Commonwealth Scientific and Industrial Research Or, European Commission – DG Eurostat, . . . Statistics Canada. (2017). *Satellite Imagery and Geospatial Data; task team report.*

ANEXO 1: EXTRACTO DE LA BITÁCORA DE TRABAJO

19 - Diciembre - 2018

Creando un usuario con permisos de root

Primero se crea el usuario:

```
sudo adduser cubeuser
```

Se asigna una contraseña:

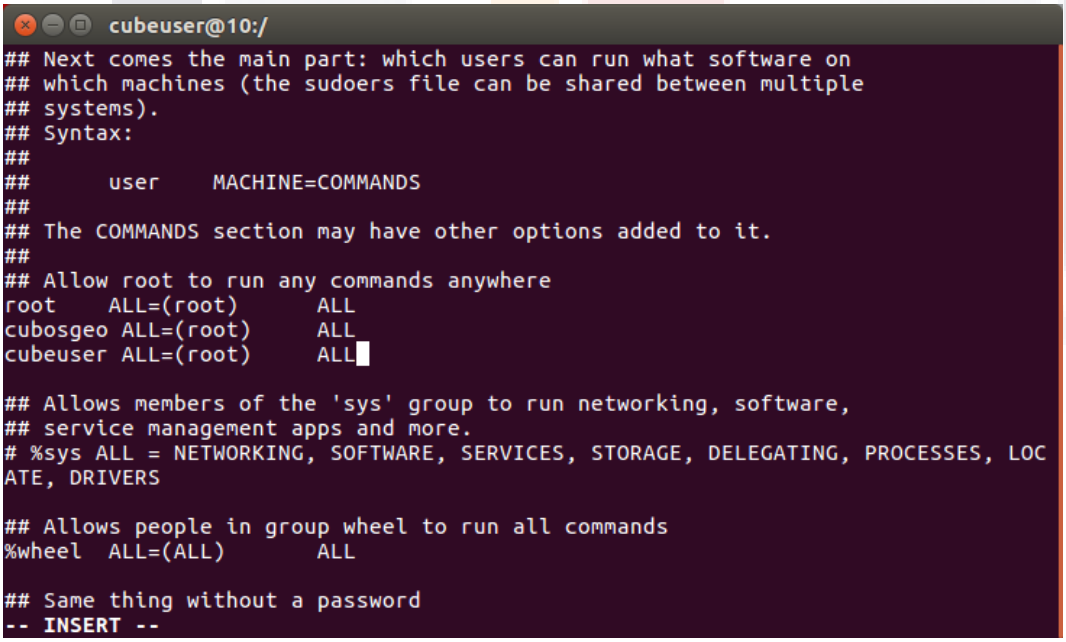
```
passwd cubeuser
```

Inegi2019

Se otorgan permisos de root modificando el archivo de sudoers por medio de visudo:

```
sudo /usr/sbin/visudo
```

Se modifica el archivo, agregando el nombre usuario en la siguiente línea, como se muestra a continuación:



Nota: La modificación se realiza con el editor vi. Se presiona *i*, para insertar texto. Para salir del modo edición es *ESC* y para guardar los cambios es *:wq*

No se puede acceder por ssh con el usuario recién creado, se debe a que Oracle Linux tiene cierta seguridad con el acceso por ssh al sistema, permitiendo que solo sea un usuario el que entre por ese medio y después cambiar de usuario haciendo uso del comando su.

07 - Enero - 2019

NOTA: Se considera la utilización del usuario cubosgeo para realizar las siguientes acciones.

Instalación wget

Después de tener algunos problemas con la conexión FTP para la transferencia del archivo de Miniconda, se dicidió probar la conexión a internet y descargar el archivo, para ello fue necesario instalar wget; el comando de instalación es el siguiente:

```
sudo yum install wget
```

Más tarde se resolvió que los problemas para el traspaso de archivos, eran debidos a falta de permisos de escritura en una carpeta.

Instalación Miniconda

Para comenzar con la instalación de Python y el cubo de datos, se utilizó Miniconda; el cual se puede descargar desde la página web del proyecto: https://www.anaconda.com/distribution/#download-section

Se descargó el instalador desde un equipo personal equipo personal y se transfirió al servidor por medio de FTP. Para iniciarlo se utilizó la siguiente instrucción:

```
bash Miniconda3-latest-Linux-x86_64.sh
```

Falló la instalación, en la lista de errores se muestra que requiere de la utilidad TAR para descomprimir archivos.

Instalación de TAR

https://www.rosehosting.com/blog/how-to-install-tar-gz-in-centos/

TAR es considerado como una de las utilidades esenciales dentro del mundo de Linux, por lo que el primer paso a realizar es instalar las “development tools”, lo que en Ubuntu se conoce como “Build-essential”, el cual es un paquete que contiene varias herramientas; para esto se ejecuta el siguiente comando en la terminal del servidor:

```
sudo yum groupinstall "Development tools"
```

Al ejecutar nuevamente el comando del punto anterior se comprobó que el problema había sido resuelto.

Iniciar Conda

Para inicializar Conda se ejecuta el siguiente comando:

```
source ~/.bashrc
```

Actualizando Conda

Una vez iniciado, se actualizó Conda con el siguiente comando:

```
conda update conda
```

Instalación del cubo de datos

Para la instalación del cubo de datos se hizo uso de la documentación de los creadores del software, la cual se encuentra en el siguiente enlace:

https://datacube-core.readthedocs.io/en/latest/ops/conda.html

Se agregó el canal de conda-forge, el cual proporciona multitud de paquetes soportados por la comunidad de usuarios de conda:

```
conda config --add channels conda-forge
```

Se creó un ambiente de Conda con python 3.6 llamado “cubeenv”, en el cual estaría instalado el cubo de datos junto con el software dependiente para su correcto funcionamiento.

El comando para crear el ambiente es el siguiente:

```
conda create --name cubeenv python=3.6 datacube
```

Para activar el ambiente se utiliza el siguiente comando:

```
source activate cubeenv
```

Para desactivar el ambiente:

```
source deactivate
```

08 - Enero - 2019

Se instalaron los paquetes de jupyter, matplotlib y scipy haciendo uso de pip; un instalador de paquetes de python.

Instalación de PostgreSQL

La instalación de postgres se realizó basado en los siguientes enlaces:

https://www.2daygeek.com/install-postgresql-on-ubuntu-centos-debian-fedora-mint-rhel-opensuse/#
http://www.techoism.com/install-postgresql-9-5-on-centosrhel-765/

Dentro del ambiente creado, se instaló postgreSQL. Lo primero que se hizo fue agregar el repositorio necesario con el siguiente comando:

```
rpm -Uvh http://yum.postgresql.org/9.5/redhat/rhel-7-x86_64/pgdg-redhat95-9.5-2.noarch.rpm
```

Se instaló el servidor de base de datos con el siguiente comando:

```
sudo yum install postgresql95-server postgresql95
```

Después se inicializa la base de datos:

```
sudo /usr/pgsql-9.5/bin/postgresql95-setup initdb
```

Para iniciar el servicio de la base de datos en el sistema se utilizó:

```
sudo systemctl start postgresql-9.5
```

Para que el servicio iniciara al arrancar el sistema, se ejecutó el comando:

```
sudo systemctl enable postgresql-9.5
```

Para asignarle una contraseña al usuario Postgres. Se inició sesión del usuario Postgres:

```
sudo su postgres
```

En seguida se ingresó a la consola de postgresql con el comando:

```
psql
```

Y al ejecutar el siguiente comando se solicita la contraseña a asignar.

```
\password postgres
```

```
*******
```

Y por último para salir de la consola de PostgreSQL:

```
\q
```

Configuración de PostgreSQL

https://unix.stackexchange.com/questions/234311/couldnt-find-postgresql-conf-pg-hba-conf-files-in-my-postgresql-installation

Se revisa y de ser necesario se modifican los siguientes archivos con el editor nano:

```
/var/lib/pgsql/9.5/data/postgresql.conf
/var/lib/pgsql/9.5/data/pg_hba.conf
```

En el primer archivo, se debe modificar la linea:

```
"Timezone = LocalTime" -> "Timezone = UTC"
```

Sin embargo, para el caso práctico no fue necesario puesto que ya estaban correctos por default

En el segundo archivo se modificó la línea:

```
# "local" is for Unix domain socket connections only
local   all             all                                     peer
```

Para quedar como sigue:

```
# "local" is for Unix domain socket connections only
local   all             all                                     md5
```

Por último, se reinició el servicio:

```
sudo systemctl restart postgresql-9.5
```

09 - Enero - 2019

Se asignó un usuario con todos los permisos asignables en Oracle Linux, las credenciales son los siguientes:

User: cubeadmin
Pass: *********

Creación del usuario dc_user

Se ingresó como el usuario Postgres

```
sudo su postgres
```

Se creó el usuario con el siguiente comando:

```
createuser --superuser dc_user
```

Se proporcionó la contrasela asignada a postgres: ********

Después se cambió la contraseña del usuario dc_user con el siguiente comando:

```
psql -c "alter user dc_user with password *******';"
```

Creación del archivo con las credenciales para la base de datos del cubo de datos

El cubo de datos requiere un archivo de configuración que apunte a la base de datos correcta y sus respectivas credenciales. El archivo debe llamarse '.datacube.conf' y estar ubicado en el directorio home del usuario local; dicho archivo se crea con el siguiente comando:

```
touch ~/.datacube.conf
```

Y el archivo debe contener lo siguiente:

```
[datacube]
db_database: datacube

# db_hostname

db_username: dc_user
db_password: [redacted]
```

El usuario y contraseña son los del usuario que recién se crearon; en caso de no especificar el hostname a donde se conectará, por default toma localhost.

Creación de la base de datos

Para la creación de la base de datos, se utiliza el siguiente comando:

```
createdb -U dc_user datacube
```

La contraseña que se pide es la de dc_user: ********
Esta base de datos se llama “datacube” y el dueño es el usuario dc_user

Inicialización del esquema de la base de datos

Para finalizar el proceso de inicialización, se ejecuta el siguiente comando para inicializar la base de datos con los esquemas y tipos de metadatos predeterminados por el cubo de datos:

```
datacube -v system init
```

Y con esto se finalizó la instalación del cubo de datos

10 - Enero - 2019

Carga de archivos

Se pasaron al servidor 654 imágenes entre Landsat 4, 5, 7 y 8 a la ruta:

```
/datos/datacube/original_data/landsat/amazon_ws
```

Se procedió al indexado de las imágenes pertenecientes únicamente a landsat 5, 7 y 8.

Indexación

En la carpeta de amazon_ws, donde se copiaron las imágenes. Con el siguiente comando se integraron en un archivo todos los archivos *".tif"* que se encontraban en la carpeta.

```
find $(pwd) -name \*.tif -type f  > Archivos_10_Ene.txt &
```

En seguida, se creo otro archivo con únicamente los archivos *".tif"* que contengan band 6, con el siguiente comando:

```
find $(pwd) -name \*band6.tif -type f  > Archivos_10_Ene_b6.txt &
```

Este último archivo tiene 654 líneas, el mismo número de imágenes que sabemos fueron transferidas al servidor.

Se abrió el archivo con el editor de texto *Sublime Text 3* y utilizando una función de la herramienta, con clic derecho y la tecla shift, se pueden editar todas la lineas al mismo tiempo; lo cual fue utilizado para borrar los nombres de los archivos y dejar únicamente a las rutas de cada archivo y se guardó el archivo con el nombre *"rutas.txt"*.

Antes de continuar con los archivos de texto, se dio de alta la definición de producto en el cubo de datos, con un archivo que define los productos para las imágenes Landsat 5, 7 y 8. El comando que se ejecuta para realizar este paso es el siguiente:

```
datacube product add ~/Documents/product_description.yaml
```

Nota: Es necesario estar en el ambiente cubeenv que se creó anteriormente

Se comprueba que el proceso terminó correctamente al obtener el siguiente resultado:

```
(cubeenv) -bash-4.2$ datacube product add ~/Documents/product_description.yaml
/datos/miniconda3/envs/cubeenv/lib/python3.6/importlib/_bootstrap.py:219: Runtim
eWarning: numpy.dtype size changed, may indicate binary incompatibility. Expecte
d 96, got 88
  return f(*args, **kwds)
Added "ls8_usgs_sr_scene"
Added "ls7_usgs_sr_scene"
Added "ls5_usgs_sr_scene"
```

Continuando con la manipulación de los archivos, se crea un archivo bash llamado *"touch_yaml.sh"* el cual contiene líneas con el siguiente formato: *"touch /ruta_imagen/nom_archivo.yaml"*; haciendo uso de las rutas que se guardaron anteriormente; se puede utilizar el mismo archivo de *"rutas.txt"*, para agregar lo que falta a las líneas con la herramienta que fueron empleados anteriormente.

Para ejecutar el bash es necesario pasarlo al servidor, se guarda en documentos y se ejecuta con la siguiente línea:

```
sh touch_yaml.sh
```

Para el siguiente paso es necesario hacer uso de un script de Python proporcionado por el equipo de Geoscience Australia y se debe crear otro archivo bash llamado *"dataset.sh"*, el cual sigue el formato: *"python* ***dataset_description.py*** */ruta_imagen/ --*

output /ruta_imagen/nom_archivo.yaml"; se puede hacer uso del archivo bash con la función de sublime text para hacer las adaptaciones.

```
sh dataset.sh
```

11 - Enero - 2019

Instalando Shapely

Al momento de ejecutar el archivo bash marcó error porque no encontró el módulo de Shapely. Se procedió a instalarlo con el siguiente comando:

```
conda install -c conda-forge shapely
```

Se volvió a ejecutar el último bash.

Se ejecutó correctamente

Continuando con el proceso de indexación, el último paso es crear un archivo bash llamado *"datacube.sh"* el cual debe estár conformado con líneas respetando el siguiente formato: *"datacube -v dataset add /ruta_imagen/nom_archivo.yaml"*

Para ejecutarlo:

```
sh datacube.sh
```

Calculando Geomediana

Se hizo una prueba con las imágenes indexadas. Para lo que se generó el archivo yaml de configuración adecuado para generar una geomediana con esas imágenes; dicho archivo fue nombrado *"celda_2018.yaml"*.

Elementos de configuración para el cálculo de la geomediana en el yaml:

- Fuentes de datos: Se tomó a consideración imágenes Landsat 5, 7 y 8.
- Región de entrada: Se referenció a un SHP con el polígono correspondiente a la CDMX; algo importante que mencionar, en este campo existe un atributo de cuadriculado (gridded), el cual es sumamente necesario mantener en falso para que se ejecute correctamente el proceso.
- Rangos de fecha: Para el caso de la prueba 2018, la fecha de inicio es: 01-01-2018; y la fecha de término es: 01-01-2019. Y pues la duración se estableció de 1 año.
- Localización: Este atributo se utiliza para asignar la dirección de salida del producto; se decidió que fuera la siguiente: ***/datos/Datacube/Productos_CDMX/2018***

Se preparó el script en python para de hacer el cálculo de la geomediana a partir de un archivo yaml de configuración, que recibe como parámetro. Además se prepara un reporte con tiempos de inicio y termino en un archivo que también se recibe como parámetro.

Se ejecutó el script y marcó algunos errores por la falta de instalación de 3 paqueterías: HDMedians, Datacube-Stats y Scikit-Image

Instalación HDMedians

```
pip install hdmedians
```

Instalación Scikit-Image

```
pip install scikit-image
```

Instalación Datacube-Stats

Se probó siguiente comando que está en la git oficial de la herramienta, sin embargo, no funcionó al momento de ejecutarlo:

```
pip install https://github.com/GeoscienceAustralia/datacube-stats/
```

Se tuvo que investigar un poco las causas y se encontró que se debía ejecutar de la siguiente manera:

```
pip install git+https://github.com/GeoscienceAustralia/datacube-stats/
```

14 -Enero - 2019

Se creó Gif con la información de la CDMX entera y otro solamente con el sur de la ciudad.

15 - Enero - 2018

Instalación de Psycopg2

```
pip install psycopg2-binary
```

Ingestión de datos

No se obtuvo un método concreto para la ingestión de datos, por lo que, para la ejecución de este paso, se utilizaron las notas que se tenían de pruebas anteriores.

Es necesaria la creación de un archivo yaml que contenga la configuración para la ingestión; se requiere un archivo por tipo de satélite. De los archivos que ya se tenían, se conserva el archivo para landsat 7 y 8; por lo que se generó el de landsat 5.

El comando para ejecutar el proceso de ingestión para landsat 7 es el siguiente:

```
datacube -v ingest -c ~/Documents/Ingestion/Archivos/ls7_lasrc_general_MX.yaml --executor multiproc 2
```

Se probó el de landsat 8 con 12 procesadores

```
datacube -v ingest -c ~/Documents/Ingestion/Archivos/ls8_lasrc_general_MX.yaml --executor multiproc 12
```

Se probó el de landsat 5 con 12 procesadores

```
datacube -v ingest -c ~/Documents/Ingestion/Archivos/ls5_lasrc_general_MX.yaml --executor multiproc 12
```

El de landsat 5, empezó a las 1:53 pm y terminó 8:50 pm. Tardó 6 H 57 M y procesó 2393 archivos.

El landsat 8, empezó a las 1:47 pm y terminó a las 8:45 pm. Tardó 6 H 58 M y procesó 1908 archivos.

El landsat 7, empezó a las 1:36 pm y terminó a las 10:50 am. Tardó 21 H 14 M y procesó 3200 archivos.

21 - Enero - 2019

Una vez que ya se contó con la ingestión de las imágenes, nos surgió la duda de cómo seleccionar las imágenes con las que se requería trabajar, las indexadas o las ingeridas en el cubo. Por lo que se consultó a los desarrolladores sobre el tema.

Con su ayuda se pudo determinar lo siguiente:

Los productos ingeridos serían nombrados:

- ls5_lasrc_general_MX
- ls7_lasrc_general_MX
- ls8_lasrc_general_MX

Mientras que los productos generados por la indexación:

- ls5_usgs_sr_scene
- ls7_usgs_sr_scene
- ls8_usgs_sr_scene

Conociendo esto, se puede modificar en los archivos YAML para la configuración de los procesos y especificar sobre qué productos se desea procesar.

Para hacer la prueba se decidió rehacer la geomediana que se hizo la semana pasada, pero ahora haciendo uso de las imágenes ingeridas. Los resultados en cuanto a la comparación de tiempo, sobrepasan lo esperado; puesto que me imaginaba que sería más rápido que con las indexadas, pero nunca creí que fuera casi del 50% la diferencia.

	Indexado	Ingestado
MAX	1:00:48	0:31:28
Prom	0:43:50	0:22:15

Se decidió poner en prueba los 2 servidores que tenemos para el cubo, uno con Ubuntu 16 y este con Oracle Linux. En el servidor de Ubuntu 16 tenemos todas las imágenes generadas en el año 2015 y 2011 únicamente; mientras que en el servidor actuar solo tenemos la historia de CDMX desde 1984 a 2018.

Como resultado de la prueba se deseaba obtener el rendimiento de ambos servidores y bajo una métrica definir cuál es el mejor. La prueba consistió en generar la geomediana del año 2015 en el área de la cuadrícula número 276 del nuevo grid con una resolución de 30m.

22 - Enero - 2019

OL7 si termino el proceso, mientras que Ubuntu 16 seguía corriendo. Algo que resultaba raro.

El log del proceso en el OL7 marcó su hora inicial y la hora de fin:
14:42:24 16:28:34

El total de tiempo fue: 01:47:50

Un tiempo despues termino el proceso de Ubuntu 16, estas fueron sus estadísticas:
14:43:28 09:43:58
El tiempo total fue de: 20:59:30

Los resultados de comparar uno contra otro parecían anormales, considerando que ambos ejecutaron el mismo código; lo único que podría variar es el número de imágenes procesadas, por lo que se dicidió contar todas las imágenes indexadas del año 2015 en cada uno de los servers. Para ello se necesitaban los path row que estaban sobre el grid 276. Estos fueron los encontrados:

- 026046
- 026047

- 025047

El servidor con Ubuntu tenía almacenadas, entre imágenes de Landsat 7 y 8, 133 imágenes; mientras que OL7 solo tenía 41.

Sin embargo OL7 procesó una tercera parte de las imágenes que UB16, por lo que si fuera proporcional, UB16 debió tardar aproximadamente 6 horas y se tardó casi las 21 horas.

Algo que es notable destacar, todas las imágenes que están indexadas en el OL7 son del path row 026047; es decir, las 41 imágenes que están en el OL7 son de ese path row.

Ubuntu en ese path row (026047) tiene 41 imágenes.

La siguiente prueba que se planeó ejecutar, consistirá en obtener una geomediana de ese path row específicamente; de esa manera, OL7 y UB16 tendrían el mismo número de imágenes a procesar, el mismo script de Python a ejecutar, los mismos núcleos de procesamiento, misma memoria RAM y solo cambiará el sistema operativo.

23 - Enero - 2019

Los tiempos obtenidos para OL7 fueron los siguientes:

- Inicio: 13:13:18
- Fin: 16:45:56
- Total: 3:32:38
- Tamaño de archivo: 504.2 Mb

Los tiempos obtenidos para UB16 fueron los siguientes:

- Inicio: Jan 22 13:14:24
- Fin: Jan 23 22:36:04

- Total: 36:22:20
- Tamaño de archivo: 503.2 Mb

ANEXO 2: SCRIPTS DE AUTOMATIZACIÓN DE LOS PROCESOS DEL CUBO DE DATOS

Proceso para la Generación de Productos

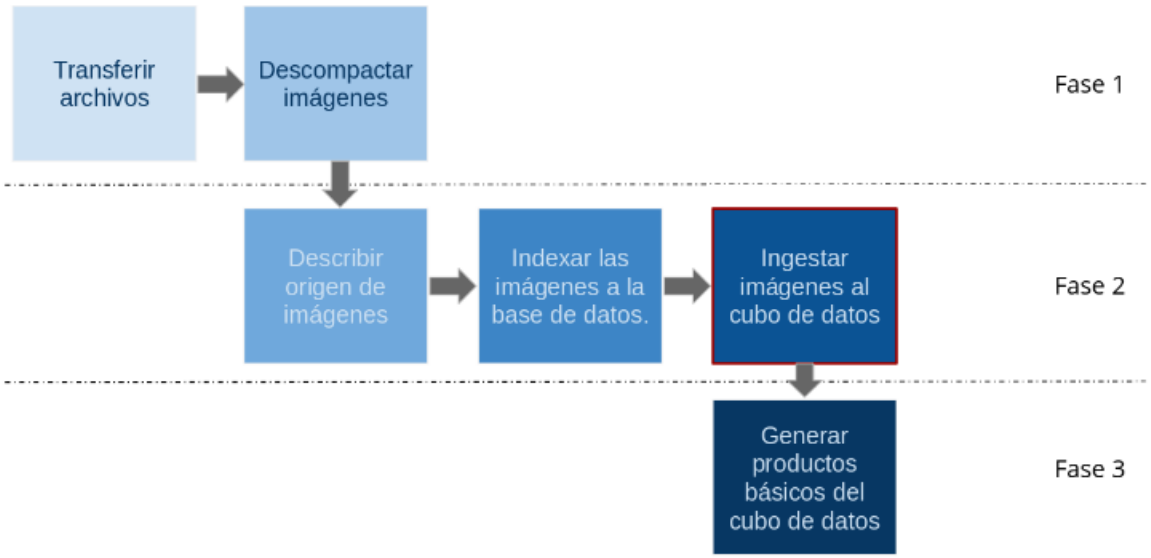


Gráfico que muestra el flujo de las fases del proceso para la generación de productos.

Fase 1

En esta fase inicial se preparan las imágenes con las que se va a procesar el producto; consiste básicamente en transferir las imágenes de la unidad de almacenamiento que contiene todo el acervo histórico de imágenes hacia la unidad de almacenamiento para procesar. Además, sobre esta fase también se contempla el proceso de descomprimir las imágenes y la revisión de su correspondiente archivo md5.

Fase 2

La segunda fase del proceso está dividida en 3 procesos principales; la descripción de imágenes, la indexación a la base de datos y la ingestión de imágenes.

1. Descripción de imágenes:

Solo es necesario ejecutar una vez este proceso, pero requiere contar con los archivos yaml que contienen la descripción de metadatos de cada sensor; de manera que cuando el cubo de datos indexe una imagen coincida con los metadatos descritos en el archivo, fungiendo como un mecanismo de seguridad para asegurarse que pueda ejecutarse correctamente.

2. Indexado a la Base de Datos:

Este proceso se ejecuta una vez por imagen, y a grandes rasgos consiste en almacenar en la base de datos la ruta donde se encuentra ubicada la imagen; haciendo destacar que se agrega un registro por cada banda espectral que contiene, siempre y cuando coincida con lo descrito en el archivo de descripción del sensor.

3. Ingestión de imágenes:

Este proceso es opcional a la hora de querer generar productos, sin embargo, el fuerte de este recurso es que permite reducir el tamaño de las imágenes con las que se va a trabajar sin perder información durante el mismo. Este proceso permite transformar las imágenes de satélite de formato GeoTiff a formato netCDF; formato que reduce el tamaño del archivo 3 veces sin perder información, de manera que si la imagen ocupaba un espacio de 1 GB, después de este proceso la misma imagen ocupará un espacio de 300 MB aproximadamente.

Fase 3

La última fase del proceso de generación de productos, básicamente consiste en crear un archivo yaml que contenga las especificaciones del producto que se desea obtener; el archivo debe contener los siguientes datos:

1. Fuente: Tipo de imágenes que se tomarán en cuenta para la generación del producto.
2. Ubicación: Donde se almacenará el resultado.
3. Región: La ubicación del archivo shape que limitar la región de trabajo.
4. Periodo del producto: Rango de fechas de la cual se espera el producto.
5. Producto: Tipo de producto que se desea obtener.
6. Salida: Se especifica la proyección y datos de salida del resultado.

Nueva Estructura del Proyecto

Grid_30m: Carpeta que contiene los archivos shape con proyección Aberst a 30 m, grid que es utilizado para la generación de productos básicos del cubo de datos.

Inventario: Carpeta que contiene los distintos inventarios que se tienen de las imágenes en el servidor y en los discos.

1. Inventario_1.csv: Este archivo contiene una lista de las imágenes contenidas en el disco 1, de los originales proporcionados por la NASA, y nos da como información el nombre del archivo, origen, ruta del archivo y tamaño en bytes.
2. Inventario_2.csv: Este archivo contiene una lista de las imágenes contenidas en el disco 2, de los originales proporcionados por la NASA, y nos da como información el nombre del archivo, origen, ruta del archivo y tamaño en bytes.
3. Inventario_3.csv: Este archivo contiene una lista de las imágenes contenidas en el disco 3, de los originales proporcionados por la NASA, y nos da como información el nombre del archivo, origen, ruta del archivo y tamaño en bytes.
4. Inventario_4.csv: Este archivo contiene una lista de las imágenes contenidas en el disco 4, de los originales proporcionados por la NASA, y nos da como información el nombre del archivo, origen, ruta del archivo y tamaño en bytes.
5. Inventario_disk.csv: Este archivo contiene una lista de las imágenes contenidas en todos los discos, de los originales proporcionados por la NASA, y nos da como información el nombre del archivo, origen, ruta del archivo y tamaño en bytes.
6. Inventario_Disco1_USGSS.csv: El archivo contiene la lista de imágenes brindadas por la USGSS, en el disco 1, como parte del segundo acervo recibido con imágenes del año 2019 y algunas bandas adicionales de años anteriores.
7. Inventario_Disco2_USGSS.csv: El archivo contiene la lista de imágenes brindadas por la USGSS, en el disco, como parte del segundo acervo recibido con imágenes del año 2019 y algunas bandas adicionales de años anteriores.
8. Inventario_server.csv: Este archivo contiene una lista de las imágenes contenidas en el disco el servidor del cubo de datos, en la partición /datos/.

9. Inventario_respaldo.csv: Es un archivo que contiene la lista de las imágenes contenidas en la ruta del respaldo; en este nuevo proceso este es el inventario en el que se basan los scripts para funcionar.
10. SensorInventory.csv: Es una lista que contiene la relación de los años con los sensores que estuvieron en uso durante ese tiempo.

Productos: Carpeta que contiene todo el software relacionado con la generación de productos básicos del cubo de datos.

Fase_1: Carpeta que contiene el software relacionado con la fase 1 de la generación de productos básicos del cubo de datos; transferencia y descompactación de imágenes.

1. copyUnzipData.py: El objetivo de la fase 1, en pocas palabras, es copiar los archivos tar las imágenes que se van a procesar, para posteriormente ser descompactadas. Este script realiza todo el proceso de la fase 1 recibiendo los siguientes parámetros:
    a. Año de las imágenes a copiar.
    b. Ruta final donde se almacenarán las imágenes ya descompactadas.
    c. Número de núcleos a utilizar.
    d. Opción de conservar archivos tar o eliminarlos. 1, para borrar; 2, para conservar.

-> Transferencia: Carpeta que contiene los scripts para realizar la transferencia de los archivos tar con las imágenes, hacia el disco de procesamiento.

1. copyData.py: Script en Python para realizar la transferencia de archivos tar hacia el disco de procesamiento. Para trabajar requiere de los siguientes parámetros:
    a. Ruta de donde se ubicarán los archivos.
    b. Año de las imágenes que desean copiar.
    c. Número de núcleos a utilizar.

2. CopyByBusqueda.py: Script con la misma funcionalidad que el anterior; aunque este script tiene la virtud de trabajar cuando no se cuenta con un inventario de las imágenes a transferir, solo que con esa virtud sacrifica tiempo y complejidad de ejecución; lo anterior es debido a que en ocasiones encuentra algunas incongruencias del número de imágenes conocidas a copiar de un año, con el número de rutas archivos que realmente encontró, en ese caso el usuario debe intervenir y arreglar el archivo de copiado. Los parámetros que recibe son:
    a. Ruta de donde se ubicarán los archivos.
    b. Año de las imágenes que desean copiar.
    c. Número de núcleos a utilizar.
    d. Opción de revisión de archivos de rutas. 0, nunca revisado; 1, ya comprobado.

-> Descompactacion: Carpeta que contiene el script para llevar a cabo el proceso de descompactación de las imágenes y almacenarlas en una ubicación dada por el usuario, de preferencia en el disco de procesamiento.

1. prepareData_OPC.py: Script en Python para ejecutar la descompactación de las imágenes de satélite, con opción de eliminar el archivo de origen. Los parámetros que requiere son:
    a. Ruta con la ubicación de las imágenes a descomprimir.
    b. Ruta final donde se almacenarán las imágenes ya descompresas.
    c. Año de las imágenes a descomprimir.
    d. Número de núcleos a utilizar.
    e. Opción de eliminar origen. 1, eliminar; 0, conservar.

Fase_2: Carpeta que contiene el software relacionado con la fase 2 de la generación de productos básicos del cubo de datos; descripción, indexación de las rutas de las imágenes a la base de datos e ingesta, ó conversión, de las imágenes a formato netCDF.

-> Descripción: Esta carpeta solo contiene los archivos yaml que describen las imágenes de los sensores landsat y sentinel, además del archivo que describe las imágenes para formato netCDF. De igual forma, se cuenta con un archivo shell que se encarga de ejecutar el proceso para el software del cubo de datos.

1. product_description.yaml: Archivo con la descripción de los sensores landsat 4, 5, 7 y 8.
2. product_definition_aea_MX_landsat.yaml: Archivo con la descripción de producto para los netCDF de los sensores landsat 4, 5, 7 y 8.
3. Product_description_sen.yaml: Archivo que contiene la descripción del sensor de Sentinel.
4. addDefProducts.sh: Script de shell que se encarga de ejecutar la sentencia para añadir la descripción de los sensores al cubo de datos, recibe como parámetro:
   a. El archivo con la definición del producto

-> Indexado: Esta carpeta contiene los scripts necesarios para realizar la indexación de las imágenes a la base de datos del cubo; los códigos están separados en 2 carpetas con el fin de dividir que es lo que se quiere indexar a la base de datos, archivos netCDF o archivos GeoTiff. Sin embargo, se cuenta con un script en Python que ejecuta todo el proceso de indexación para cualquiera de los 2 casos.

1. Indexing_all.py: Este es el script que indexa tanto TIFs como archivos netCDF a la base de datos del cubo; para ejecutarse correctamente requiere de los siguientes parámetros:
   a. Ruta donde se encuentran los archivos a indexar.
   b. Tipo de archivo que se va a indexar. 1, para TIFs; 2, para netCDF.
   c. En caso de ser TIFs, el tercer parámetro es el número de núcleos a utilizar.

-> netCDF

1. indexingNETCDF.sh: Es un script en shell que ejecuta la indexación de archivos netCDF a la base de datos del cubo, el único parámetro que requiere es:
    a. Ruta donde se encuentran los archivos netCDF a indexar.

-> TIFs

1. t_gen rutas.sh: Es un script en Python que se encarga de obtener la ruta de las bandas de cada imagen a indexar y las guarda en un archivo txt. Los parámetros que requiere son:
    a. Ruta donde se encuentran las carpetas de las imágenes.
    b. Ruta del txt donde se guardará el resultado.

2. t_genYAML.py: Es un script en python que se encarga de crear un archivo *yaml* por cada imagen, nombrado de la misma forma que la carpeta que contiene la imagen. El parámetro que requiere es:
    a. La ruta del archivo txt generado por t_genRutas.sh.

3. t_genDataset.py: Este script es el encargado de llenar los archivos yaml creados por *t_genYAML.py*; escribiendo sobre los archivos las características de las imágenes a indexar. Los parámetros que requiere:
    a. La ruta del archivo txt generado por t_genRutas.sh.
    b. El número de núcleos a utilizar.

4. dataset_description.py: Script de Python compartido por el equipo técnico de Geoscience Australia; con el fin de completar los archivos yaml para la indexación de los archivos geoTIFF. Este script es requerido por el descrito en el punto anterior.

5. t_genDatacube.py: Script de python para añadir las imágenes a la base de datos, haciendo uso de los yaml creados por los scripts anteriores. El parámetro que requiere es:
    a. La ruta del archivo txt generado por t_genRutas.sh.
    b. El número de núcleos a utilizar.

-> Ingestión: Esta carpeta contiene los archivos necesarios para ejecutar el proceso de ingestión; la conversión de las imágenes de formato GeoTiff a NetCDF. Es un archivo por sensor; en nuestro caso son archivos yaml con las características necesarias para los sensores landsat 4, 5, 7 y 8.

1. ingestion_ls4_aea_MX_final.yaml: Es un archivo yaml de descripción para de las imágenes TIF de landsat 4, con el cual se hace una especie de mapeo para poder regenerar la imágen en formato netCDF, de manera que se reduce el tamaño del archivo sin perder información de la imágen.

2. ingestion_ls5_aea_MX_final.yaml: Es un archivo yaml de descripción para de las imágenes TIF de landsat 5, con el cual se hace una especie de mapeo para poder regenerar la imágen en formato netCDF, de manera que se reduce el tamaño del archivo sin perder información de la imágen.

3. ingestion_ls7_aea_MX_final.yaml: Es un archivo yaml de descripción para de las imágenes TIF de landsat 7, con el cual se hace una especie de mapeo para poder regenerar la imágen en formato netCDF, de manera que se reduce el tamaño del archivo sin perder información de la imágen.

4. ingestion_ls8_aea_MX_final.yaml: Es un archivo yaml de descripción para de las imágenes TIF de landsat 8, con el cual se hace una especie de mapeo para poder regenerar la imágen en formato netCDF, de manera que se reduce el tamaño del archivo sin perder información de la imágen.

5. Ingest.sh: Es un script en shell que ejecuta la función del cubo de datos para realizar el proceso de ingestión a partir de las imágenes indexadas en la base de datos. Los parámetros que requiere son:
    a. El archivo que contiene los parámetros de ingestión.
    b. El número de núcleos a utilizar.

Fase_3: Carpeta que contiene los scripts necesarios para llevar a cabo el último paso para la generación de los productos de un determinado año.

1. createProducts.py: Es un script en python que genera los productos definidos en varios archivos yaml previamente creados y referenciados en otros archivos que contienen sus ubicaciones. Los parámetros que requiere son:
    a. La ruta del archivos de texto con las rutas de los archivos yaml.
    b. La ruta donde se almacenarán los productos.

2. createProducts_ind.py: Es un script en python que genera los productos definidos en un único archivo yaml, a diferencia del script anterior. Los parámetros que recibe son:
    a. La ruta de la ubicación del archivo yaml.
    b. La ruta donde se almacenarán los productos generados.

3. water_classifier_test.py: Es un script de python compartido por Geoscience Australia, su función es la de apoyar con la generación del producto WOFS, que es el que analiza los cuerpos de agua. No requiere parámetros, solo debe estar ubicado en la misma ruta que el archivo que lo manda llamar, en este caso, cualquiera de los 2 anteriores.
4. genProducts.py: Este script es el que automatiza en su totalidad la creación de los productos; crea los archivos yaml, genera los archivos con las rutas de los archivos yaml, y genera el script en shell que se encarga de ejecutar todo lo

anterior. Para que este script funcione correctamente, debe recibir los siguientes parámetros:

a. Año del que se procesarán productos.
b. Ruta donde se almacenarán los productos.
c. Tipo de archivo con el que se va a trabajar. 1, TIFs; 2, netCDF.
d. Ruta donde se ubican los archivos shp para delimitar el área de donde se obtendrán los productos.
e. Número de núcleos a utilizar.
f. Con landsat 7 1. NO; 2. SI

-> YAML_generator: Esta carpeta contiene varios scripts que tienen como fin generar archivos yaml para la generación de productos, en base a lo que el usuario requiera. Es importante destacar que los archivos yaml, producto de estos scripts, se almacenan en la ruta: /TMP_Resultados/Productos/YAML/

1. nc_YAMLgenerator_gm.py: Este es un script de python que genera los archivos yaml para la generación de geomediana a partir de archivos netCDF. Es importante destacar que para hacer un uso correcto de este script es necesario entrar al código y hacer las modificaciones pertinentes sobre lo que se va a guardar en cada archivo. Los parámetros que requiere son:
    a. Año del que se generarán productos.
    b. Ruta donde se almacenarán los productos.

2. nc_YAMLgenerator_rest.py: Es un script similar al anterior, salvo por la diferencia de que esté crea los archivos yaml para generar el resto de los productos indicadores (excepto la geomediana). Los parametros son:
    a. Año del que se generarán productos.
    b. Ruta donde se almacenarán los productos.

3. t_YAMLgenerator_gm.py: Este es un script de python que genera los archivos yaml para la generación de geomediana a partir de archivos TIF. Es importante destacar que para hacer un uso correcto de este script es necesario entrar al código y hacer las modificaciones pertinentes sobre lo que se va a guardar en cada archivo. Los parámetros que requiere son:
    a. Año del que se generarán productos.
    b. Ruta donde se almacenarán los productos.

4. t_YAMLgenerator_rest.py: Es un script similar al anterior, salvo por la diferencia de que esté crea los archivos yaml para generar el resto de los productos indicadores (excepto la geomediana). Los parámetros son:
    a. Año del que se generarán productos.
    b. Ruta donde se almacenarán los productos.

5. yamlGenerator.py: Este es un script más elaborado, pero tiene el fin de facilitar la generación del archivo yaml; en este script ya no es necesario que el usuario navegue por el código para ajustarlo de acuerdo a las necesidades que tenga. De igual manera, el usuario puede dar por parámetro si se va a trabajar con archivos netCDF o con TIF. Los parámetros que requiere para funcionar correctamente son:
    a. Año del que se generarán productos.
    b. Ruta donde se almacenarán los productos.
    c. Tipo de archivo a trabajar. 1, TIF; 2, netCDF.
    d. Ruta del directorio con los archivos shp a procesar. Estos archivos delimitan la zona de donde generar los productos.
    e. Ruta donde se almacenarán los yaml. Aquí el usuario puede elegir dónde guardarlos.
    f. Lista con los productos a generar (Revisar /Productos/Fase_3/genProducts.py para ver relacion de lista).

TMP_Resultados: Esta carpeta contiene todos los archivos txt y sh temporales, es decir que su uso es variante; el hecho de que sean temporales no quiere decir que se eliminarán como los temporales de sistema, sin embargo, son archivos cuyo contenido puede llegar a cambiar por cada corrida de cada script. Algunas de las salidas de los scripts de la carpeta Útiles (El punto siguiente) tienen su salida de resultados a algún(os) archivos que se almacenan aquí. Además, hay carpetas internas que ayudan a organizar de mejor forma la estructura de guardado de los archivos.

1. 2Compact: En esta carpeta se almacenan los archivos de los productos básicos del cubo de datos que se van a compartir, puede almacenar los archivos .zip ó los archivos de los productos por separado. Depende del usuario vaciar esta carpeta.

2. 2Copy: En el punto de la carpeta productos que se vió en el punto anterior, se mencionó de un script que puede funcionar sin la necesidad de contar con un inventario de imágenes; es en esta carpeta donde se almacenan todos aquellos elementos que requiere para funcionar correctamente.

3. Indexados: Durante el proceso de generación de productos, hay una etapa de indexado a la base de datos, en esta carpeta se guardan aquellos datos que requiere para realizar dicha acción.

4. Paralelizados: Esta carpeta tiene como objetivo contener todos lo archivos shell requeridos para “paralelizar” un proceso.

5. tmpFase3: Los archivos yaml de la generación de productos y los archivos con las rutas de esos archivos, se almacenan en esta carpeta. Así como también podemos encontrar los logs de la fase 3 del proceso de generación de productos.

Útiles: Carpeta que contiene scripts en python que funcionan como herramientas para el usuario; varias de estas herramientas son utilizadas por otros scripts en la generación de productos básicos del cubo de datos. Además, el usuario puede obtener información general del estado del cubo a partir de la combinación de estas herramientas.

1. c_CountFiles.py: Este script fue diseñado con el fin de hacer una verificación de la extracción de los archivos de imágenes compresas en TAR; donde si la carpeta contenía el número de archivos esperados estaba correcta.
   El script consiste básicamente en contar el número de archivos dentro de una carpeta de un grupo de directorios ubicados en una misma ruta; generando así el archivo TMP_Resultados/descompactResult.txt, donde se listarán aquellas carpetas que no coincidieran con el número de archivos esperados por el usuario.
   a. La ruta donde se ubican los directorios a analizar.
   b. Número de archivos que se esperan por carpeta.

2. c_Img&Sensor_PathRow.py: Script en python que genera un archivo csv con relación de path rows y cuantas imágenes existen por sensor. El resultado lo almacena en TMP_Resultados/CalculoImagenes.csv, y no recibe parámetro para su correcta ejecución.

3. c_Img&Sensor_PathRow_especifico.py: Script en python que genera un archivo csv que contiene el listado de imágenes de un año especificado por el usuario; el número de imágenes se muestran separadas por path row y por el sensor que la capturó. El resultado lo almacena en TMP_Resultados/CalculoImagenes.csv
   a. Año para filtrar.

4. c_Img&Tam_Anual.py: Script en python que guarda en un archivo el número de imágenes para un año específico, el espacio que ocupan en disco en bytes y el número de imágenes que pertenecen al año, separadas por sensor. El archivo generado se almacena en TMP_Resultados/c_IMGtam_anual.txt
   a. Año a buscar.

5. e_executeSH.py: Es un script en python que consiste en ejecutar todos los archivos shell que se ubiquen en la carpeta TMP_Resultados/Paralelizados/, de manera que simule una ejecución en paralelo de todos ellos, ya que los ejecuta en segundo plano.
   a. Ruta de la carpeta Paralelizados. Se pide por parámetro debido a que la ubicación varía dependiendo de donde mande ejecutar el script.

6. e_Paralelizar.py: Este script tiene la función de dividir un archivo shell con varias instrucciones similares en un determinado número de archivos; estos archivos son los que ejecuta el script anterior. Los archivos resultantes de este proceso se almacenan en la carpeta TMP_Resultados/Paralelizados.
   a. La ruta del archivo shell a dividir.
   b. El número de archivos que se desean obtener.

7. getInventory.py: Este script de python genera los inventarios de imágenes ubicadas en una determinada ubicación. Este inventario consiste en listar las imágenes ubicadas en una ruta y separarlas por los siguientes datos: nombre del archivo, origen de la imágen, ruta donde se ubica el archivo y el espacio que ocupa en disco.
   a. Ruta donde se ubican las imágenes a inventariar.
   b. Origen de esas imágenes.

8. ingestedSeparator.py: Es un script de python que fue creado con la finalidad de separar los archivos ingeridos por el cubo de datos, de un año especifico, y moverlos a la ruta final de este tipo de archivos.
   a. Ruta donde se ubican los archivos ingeridos.
   b. Año que se está buscando.

9. p_checkProductProcess.py: Este script tiene una función muy simple, contar el número de archivos de productos generados y clasificarlos según su tipo, de manera

que al darle ese número al usuario pueda darse una idea del avance que se tiene en la generación del producto.

a. La ruta de la ubicación de los productos.

10. p_contSizeProducts.py: Es un script muy similar al anterior, con la diferencia de que este no menciona el progreso, sino nos da el tamaño total de los archivos como dato adicional.

a. La ruta de la ubicación de los productos.

11. p_joinProducts.py: Los productos generados por el cubo de datos, los almacena a todos juntos en una misma ubicación, entonces para poder obtener únicamente los archivos correspondientes a un tipo de producto fue creado este script. Además, este mismo script es capaz de comprimirlos si es que fuera necesario compartirlos por algún medio.

a. La ruta de la ubicación de los productos.

12. u_checkMD5.py: Este script en Python compara el md5 de los archivos originales con los copiados en una ruta distinta; es necesario que los archivos *.md5* estén junto a los archivos originales.

a. Ruta donde se encuentran los archivos copiados.

13. u_checkMD5_ind.py: Idéntico al anterior salvo por una diferencia, este script compara los md5 de un archivo individual.

a. La ruta con la ubicación del archivo md5 a comparar.

14. u_cmpFiles.py: Script en Python que compara las líneas de 2 archivos dados por parámetro, y escribe las que no coinciden en un archivo txt ubicado en TMP_Resultados/res_cmpFiles.txt.

a. Archivo 1 para comparar.
b. Archivo 2 para comparar.

15. u_copyIMG_pathrow.py: Script en Python que copia un conjunto de imágenes de un path row y año específico a una ruta dada por el usuario. Es importante mencionar que este script copia las imágenes en formato TIF sin comprimir.
    a. Path row de las imágenes a copiar.
    b. Año de las imágenes a copiar.
    c. donde se encuentran las imágenes a copiar
    d. Ruta a donde se van a copiar.

16. u_ubicarIMG_PathRow.py: Script en Python que nos muestra el número de imágenes que se ubican en un path row específico, dado un determinado año.
    a. Año que se desea buscar.
    b. Path Row a encontrar.

17. u_DescompactFiles.py: Este es un script sencillo en Python que simplemente descomprime todos los archivos tar de una ubicación a una ruta especificada por el usuario. El script permite “paralelizar” el proceso y ejecutarlo en un cierto número de núcleos para acelerar el proceso.
    a. La ruta donde se encuentran los archivos tar.
    b. La ruta donde se almacenarán los archivos descomprimidos.
    c. El número de núcleos a utilizar.
    d. La opción de borrar los tar de origen o conservarlos. 1, borrar; 2, conservar.

18. u_getFilesBYyear.py: Este script en python lista todos los archivos de un año, especificado por el usuario, que se encuentren en el inventario de imágenes. El resultado lo guarda en TMP_Resultados/imagenes.txt.
    a. El año a analizar.

19. u_copyIMG_pathRow.py: Este script en python, fue creado con la intención de ubicar las imágenes de un año especifico en un path row determinado.
    a. Ruta donde se encuentran las imágenes.
    b. Path row que se busca.
    c. Año de las imágenes que se busca.

Logros Alcanzados

Una vez terminada la tarea de reestructurar y codificar las tareas pendientes para el proceso de generación de productos básicos del cubo de datos, conseguimos un 85% de automatización; considerando como medición el número de intervenciones que tenía el usuario sobre el proceso. De acuerdo con las mediciones que se tomaron antes de la reestructuración, el usuario tenía 20 intervenciones para cubrir todo el proceso de la generación de productos.

Ahora, con la nueva estructura y códigos de automatización que se implementaron, el usuario puede generar uno o varios productos con solo 3 intervenciones. Esto permite al usuario liberar su carga de trabajo; monitoreando el avance del proceso y sus resultados para intervenir cuando sea necesario.