

Universidad Tecnológica de la Costa



División de Ingeniería Ciencia y Tecnología

**Implementación de Cadena de Descarga Automática de
Datos Satelitales (Parte 2)**

MEMORIA

Que para obtener el título de:

Ingeniería en Tecnologías de la Información y Comunicación

Presenta:

Oscar Omar Pineda García

Asesor:

MCE Héctor Hugo Domínguez Jaime

Dedicatorias

Este proyecto se lo dedico principalmente a mis padres quienes me han dado su apoyo incondicional para poder seguir adelante, ya que sin ellos no hubiera podido terminar mi carrera. De igual manera se lo dedico a mi hermana quien siempre me ayudo a poder cumplir mis objetivos y ser alguien responsable con la escuela.

Agradecimientos

Gracias a la Universidad Tecnológica de la Costa por haberme permitido estudiar en su institución educativa, la cual es la máxima casa de estudios de la zona, así como a los maestros de la carrera de Tecnologías de la Información y la Comunicación Área Sistemas Informáticos quienes me enseñaron lo necesario para ser alguien competente en mi área de trabajo.

Gracias a mis compañeros y amigos por haber estado con se les necesitaba y por su gran apoyo incondicional.

Índice

Capítulo 1. Planteamiento del problema.....	1
1.1 Antecedentes	1
1.2 Objetivos	2
1.2.1 Objetivos específicos.....	2
1.3 Justificación.....	3
1.4 Limitaciones	4
Capítulo 2. Marco Teórico.....	5
2.1 Percepción remota.....	5
2.1.1 sensores.....	5
2.2 Protocolo HTTP	9
2.2.1 Peticiones GET y POST.	10
2.3 scripts.....	11
2.3.1 Python.....	11
2.4 Diagrama de Gantt.	12
2.5 Scrum	13
2.6 Técnicas de Recolección de datos.....	14
Capítulo 3. Metodología	15
3.1 Planeación	15
3.1.1 Gráfica de Gantt.....	15
3.1.2 Metodología de desarrollo de software.	16
3.2 Fase de Análisis.....	17
3.2.1 Técnica de recolección de datos.	17
3.2.2 Requerimientos funcionales y no funcionales.	19
3.3 Fase de Diseño	19
3.4 Construcción de ingeniería.....	23
3.4.1 Codificación.....	23
3.4.2 Pruebas del Script.....	27
Capítulo 4. Resultados	32

Capítulo 5. Conclusiones	35
Referencias.....	38
Anexos	39
Redacción de procesos de L2	39
Notas de Simbología.....	39
Primera parte↴.....	39
Función <i>main</i> ○	40
Script <i>descarga</i> ○	41
Función <i>proDescarga</i> ○	42
Script <i>lectorNC</i> función <i>comprobacionNC</i> ○	42
Redacción de Procesos de L3	43
Primera parte.....	43
Función “ <i>main</i> ”.....	43
Script <i>formato_juliano</i>	44
Script <i>generador</i>	44
Script <i>descargador</i>	45
Script <i>descarga</i>	45
Script <i>lectorNC</i>	47

Índice de tablas y/o figuras

Figura 1. Satélite.	5
Figura 2. Toma con el sensor MODIS.	8
Figura 3. HTTP.	10
Figura 4. Ejemplo de Grafica de Gantt.	13
Figura 5. Grafica de Gantt.	15
Figura 6. Agenda.....	15
Figura 7. Metodología SCRUM.....	17
Figura 8. Ejecución Principal de Descarga L2.....	20
Figura 9. Función Main de Descarga L2.	21
Figura 10. Función encargada de descargar de L2.	22

Figura 11. Función Pro_Descargas de descargas L2.	22
Figura 12. Función para leer los NC de descargas L2.	23
Figura 13. Función Principal de descarga L2.	24
Figura 14. Función main de descarga L2.	25
Figura 15. Archivo de Configuración de descarga L2.	26
Figura 16. Archivo de configuración de Procesamiento L2.	26
Figura 17. Función que se encarga de procesar.	27
Figura 18. Lista de los NC.	28
Figura 19. Error de Ruta de Archivo no encontrada.	28
Figura 20. Ejemplo de Variables Extraídas de los NC.	29
Figura 21. Comprobación para saber si devolvió la ruta del XML.	30
Figura 22. Ejemplo de XML Creado.	30
Figura 23. Prueba en consola de Procesamiento terminada.	31
Figura 24. Archivo Procesado.	31
Figura 25. Archivo de configuración para L2.	32
Figura 26. Descargando Archivos.	33
Figura 27. Archivos NC dentro del directorio.	33
Figura 28. Imágenes L2 por procesar.	34
Figura 29. NC procesado.	34
Figura 30. Muestra de un NC de L2 convertido a L3.	35

Resumen

El Centro Nayarita de innovación y Transferencia de Tecnología (CENIT) en el laboratorio de percepción remota (PERSEO) se lleva a cabo investigaciones relacionadas con el análisis de clorofila de cuerpos de agua, además de recopilación de información oceanográfica y costera, etc. Todo esto mediante la descarga de datos satelitales de diferentes sensores lanzados por la NASA como son los sensores MODIS Aqua, MODIS Terra y VIIRS. Dicha descarga masiva de información puede resultar tediosa y cansada además de consumir demasiado tiempo de trabajo importante que se podría emplear en otras etapas de la investigación. De igual manera la forma actual de obtener los datos es ambigua e incluso puede ser peligrosa cuando se usan aceleradores de descarga, por eso mismo se optó por desarrollar scripts de descargas automatizadas de productos nivel 2 (L2) y nivel 3 (L3) de los sensores antes mencionados, tales scripts permitirán de una manera sencilla descargar cientos de archivos necesarios para las investigaciones pertinentes del área de PERSEO, optimizando así el tiempo de trabajo. Dicho proyecto se realizó con la metodología de SCRUM debido al entorno complejo, donde se necesita obtener resultados periódicamente y los requisitos son cambiantes o pocos definidos, donde la innovación, la competitividad la flexibilidad y la productividad son fundamentales. Con ello se concluye con scripts fáciles de utilizar para la descarga automatizada de archivos satelitales de extensión NC.

Capítulo 1. Planteamiento del problema

1.1 Antecedentes

El Centro Nayarita de Innovación y Transferencia de Tecnología ubicado en la Ciudad del Conocimiento en Tepic Nayarit es un centro de investigación de alto nivel el cual pone a disposición de la sociedad las herramientas científicas y tecnológicas con las que cuenta para así generar alianzas con instituciones de educación superior y otros centros de investigación del país. Todo con la finalidad de fortalecer la docencia y fomentar la investigación e innovación.

CENIT nace de la alianza entre CONACYT y la Universidad Autónoma de Nayarit, siendo una institución dedicada a la investigación CENIT recibe a practicantes y tesis para que desarrollen proyectos relacionados con la innovación.

Dentro de las actividades que se desarrollan en CENIT está el monitoreo de mares a través de la percepción remota o tele detección satelital, el centro de investigación no cuenta con satélites propios por lo que hace uso de una plataforma web llamada OCEAN COLOR en la que se encuentra una serie de archivos con información real y veraz sobre distintos elementos que se evalúan en el monitoreo. Para hacer uso de estos archivos es necesario descargarlos, el problema a resolver en el desarrollo de este proyecto nace debido a que esas descargas son de forma individual y es necesario llevar a cabo descargas masivas de archivos, ya sea por los tipos de estudios que se llevan a cabo, por años completos o ciertos periodos de tiempo.

Actualmente no se cuenta con una herramienta que realice las descargas de forma automática por lo que es necesario desarrollar un algoritmo que descargue los archivos directamente de OCEAN COLOR y los almacene en un servidor con el que cuenta CENIT para posterior a su descarga sean procesados lo cual consiste en una conversión de archivos NC a una imagen del

mundo en la que se observan una variedad de colores y en base a los colores los investigadores puedan interpretar los resultados que se encuentran en los archivos.

1.2 Objetivos

Desarrollar scripts para la descarga automatizada de datos satelitales de la NASA de productos nivel 2 y 3 tipo Mapped de los sensores MODIS-Aqua, MODIS-Terra y VIIRS. Esto con la finalidad de una mejora en la obtención de los datos para fines de investigación.

1.2.1 Objetivos específicos

- Obtener los requerimientos mediante reuniones.
- Diseñar las ventanas de función.
- Desarrollar los scripts mediante el lenguaje de programación de Python versión 2.7.
- Realización de pruebas a los scripts.
- Creación de manuales de usuario.

1.3 Justificación

En CENIT en el área de PERSEO se realizan investigaciones que contribuyen al estudio y monitoreo de aguas costeras, entre sus objetivos está el de prevenir a la población de posibles daños a la salud por posibles crecimientos anómalos de microorganismos dañinos; todo esto mediante el análisis de datos oceanográficos compartidos por la NASA, los cuales son tomados por diferentes sensores, entre los más relevantes para el área de investigación de PERSEO se encuentran MODIS-Aqua, MODIS-Terra y VIIRS.

Estos datos son descargados de manera frecuente de una manera ambigua y tardada. Por tal motivo los scripts permitirán al investigador o cualquier usuario final de una manera fácil editar un archivo de configuración con los datos correctos para la descarga masiva de archivos con extensión NC sin importar la cantidad total que se requieran.

Para poder realizar los scripts fue necesario conocer las políticas de seguridad que la NASA ha implementado en sus servidores donde almacenan sus datos que comparten, para así obtener una mejor comunicación entre el script y los servidores; pudiendo descargar miles de datos sin problemas, además de que se requirió de la ingeniería social para poder entender cómo funcionan sus páginas y poder nosotros interactuar de una manera eficaz con ellas.

Todo el trabajo realizado tendrá un gran impacto dentro del centro de investigación haciendo que se obtenga conocimientos sobre los cuerpos de agua que se analicen de una forma más rápida.

1.4 Limitaciones

En este proyecto solo nos enfocaremos en la descarga de datos satelitales de tipo Mapped de los sensores MODIS-Aqua, MODIS-Terra y VIIRS de nivel 3 y de productos nivel 2 por su gran complejidad extensión que estos representan. Por tal motivo para poder ser cumplido los objetivos antes propuestos el equipo de trabajo estará constituido por 3 personas.

Se realizarán las validaciones correspondientes en los scripts de descarga para que los usuarios finales cometan el mínimo error al interactuar con él. Se realizarán los manuales de usuario correspondientes para cada script y se realizarán las pruebas para cerciorarse de que funcionen como se cabe esperar.

Capítulo 2. Marco Teórico

2.1 Percepción remota

Desde los comienzos de los tiempos el hombre ha buscado conocer gráficamente su entorno y eso lo ha llevado a grandes avances científicos, hoy día existen se ha desarrollado la tecnología satelital la cual ha facilitado la obtención de información de una manera más amplia y con continuidad en el tiempo. Los satélites de acuerdo Murillo Sandoval & Carbonell González (2012) son vehículos colocados en órbita alrededor de la tierra. Y cuentan con la ventaja de recolectar información regular de la superficie terrestre mediante sensores.

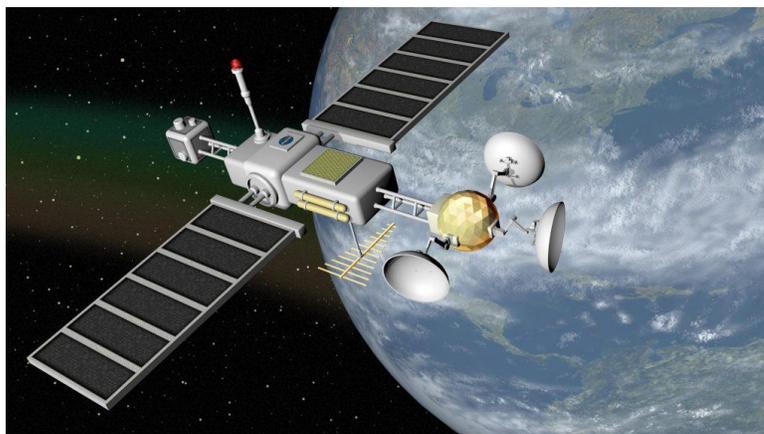


Figura 1. Satélite.

2.1.1 sensores.

Existen diferentes tipos de sensores según la región del espectro en el que trabajen, y se clasifican en: microondas, visibles e infrarrojos. Y según el tipo de energía usado, ya sea de tipo activos y pasivos. Lo importante señalar son los sensores pasivos ya que estos son aquellos que se limitan a recibir energía proveniente de un foco exterior a ellos y los activos cuando son capaces de emitir su propio haz de energía, siendo más flexibles por no depender tanto de las condiciones exteriores al sistema sensor-tierra lo cual permite que estén obteniendo datos tanto

de día como de noche (Olmo, s.f.). Entre las cosas que pueden ser medidas con los sensores se encuentran el color, temperatura, pendiente, altura y rugosidad.

Los sensores cuentan con resoluciones las cuales se pueden definir como la habilidad de registrar información de detalle, discriminándola. Existen cuatro manifestaciones de resolución las cuales son: espacial, espectral, radiométrica y temporal.

La resolución espacial hace referencia al objeto de tamaño más pequeño que puede ser distinguido en una imagen producida por un sensor (Olmo, s.f.), se mide en unidades de longitud sobre el terreno y depende de la apertura focal, la cámara y de su altura sobre la superficie. La resolución espectral indica el número y anchura de las bandas espectrales que pueden detectar un sensor, los mejores sensores son los que cuentan con más bandas y son más estrechas (Murillo Sandoval & Carbonell González, 2012). Hablando de la resolución temporal podemos referirnos a la periodicidad con la que el sensor adquiere datos del mismo sector sobre la superficie terrestre (Olmo, s.f.). El tener altas resoluciones temporales es muy importante cuando se intenta monitorear eventos que cambian en periodos relativamente cortos como por ejemplo la calidad del agua. Y por último esta la resolución radiométrica la cual hace mención a la sensibilidad del sensor o en otras palabras la capacidad de detectar la radiación espectral que recibe.

2.1.1.1 MODIS.

El sensor MODIS se encuentra a bordo de los satélites Terra o EOS-AM y Aqua o EOS-PM, la órbita de estos son helio-sincrónica y cuasi-polar con una inclinación de 98.2° y 98°. Terra está programada para pasar de norte a sur cruzando por el ecuador a las 10:30 am en su órbita descendente mientras que Aqua pasa de sur a norte sobre el ecuador a las 1:30 pm. Ambas plataformas monitorean la totalidad de la superficie terrestre cada 1 o 2 días dependiendo de la

latitud. Cuenta con una alta sensibilidad radiométrica en 36 bandas espectrales, en un rango de longitud de onda que va de los 0.4 μm a los 14.4 μm . MODIS posee una alta calidad geométrica que permite el monitoreo preciso de las alteraciones de la superficie terrestre. (Mas, Jean-Francois, 2011)

Los productos MODIS son divididos en cinco niveles (0 a 4) en función del grado de procesamiento realizado:

- Nivel 0: son los datos brutos sin ningún tratamiento, solo los artefactos de comunicación han sido eliminados, como por ejemplo los paquetes de sincronización y los encabezados de comunicación entre el satélite y la estación de recepción.
- Nivel 1: son los datos de geolocalización que contiene coordenadas geodésicas, información sobre la elevación del terreno, máscara de tierra/agua, Angulo de elevación, cenit y azimut del satélite y del sol.
- Nivel 1A: son los productos utilizados para la geolocalización, la calibración y el procesamiento. Contienen los niveles de radiación de las 36 bandas, junto con datos auxiliares del sensor y del satélite como coeficientes de calibración radiométrica y corrección geométrica, parámetros de georreferenciación (efemérides de la plataforma).
- Nivel 1B: son los productos que contienen las radiancias calibradas y con geolocalización para las 36 bandas generadas por el nivel 1A. Es por lo tanto una imagen radiométricamente corregida y calibrada a unidades físicas. Datos adicionales incluyen estimaciones de calidad, del error y datos de calibración.

- Nivel L2: son los productos que contienen variables geofísicas. Estos productos se generan a partir del producto nivel 1B aplicando correcciones atmosféricas y algoritmos bio-ópticos. Estos productos generados pueden ser almacenados en gránulos, que corresponden a 5 minutos de datos colectados por MODIS, de tal forma que cada gránulo es de aproximadamente 2,340 x 2,330 km.
- Nivel 3: son productos de valor agregado derivados de variables geofísicas mapeadas. En algunos casos, se realiza un re muestreo a una menor resolución espacial que las bandas originales y se elaboran compuestos de intervalos de tiempo establecidos (1, 8, 16 y 30 días).
- Nivel 4: son los productos generados por la incorporación de los datos MODIS en modelos para así estimar variables geofísicas. Estos productos se generan usando productos nivel L2 y datos auxiliares.

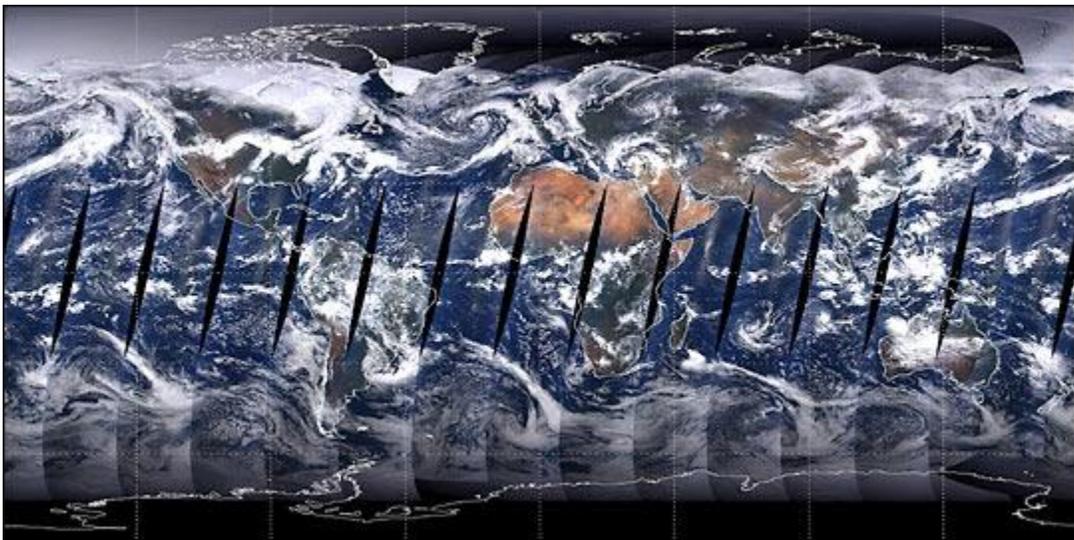


Figura 2. Toma con el sensor MODIS.

2.1.1.2 VIIRS.

Es el sucesor de MODIS y su cobertura espectral va de los 412 nm a los 12 micrones. Cuenta con 22 bandas y las imágenes tienen una resolución de ~375 m al nadir en 5 bandas y de ~750 m al nadir en las bandas restantes. Cuenta con una cobertura global completa, productos de nubes, propiedades de aerosoles, tierra y océanos. La cobertura espectral es ligeramente menor, las bandas infrarrojas son más largas para CO₂, el sistema de telescopio rotatorio permite un mejor control de la luz, la órbita es más alta, lo que permite una cobertura global completa en un día. La calidad radiométrica y calidad espectral es similar a MODIS con datos de 12 bit, las calibraciones a bordo del satélite también son parecidas a las de MODIS, el impacto en el índice de vegetación es mínimo. La reducción de bandas de 36 en MODIS a 22 Aplicaciones del sensor MODIS para el monitoreo del territorio a 22 en VIIRS se compensa en parte debido a las 7 bandas de amplificación dual en VIIRS que pueden ser usadas tanto para obtener datos de cobertura terrestre como de superficies oceánicas, ya que cuenta con un sistema que permite ajustar las mediciones del sensor de acuerdo a la superficie observada. (Mas, Jean-Francois, 2011)

2.2 Protocolo HTTP

Hablando sobre la interacción que se hace entre los scripts y los servidores de la NASA es necesario que se adopten buenas prácticas de comunicación para que los servidores nos permitan el libre acceso a su información y para eso está el protocolo de transferencia de Hipertexto el cual es de tipo cliente- servidor. Cada vez que se activa este protocolo se sigue un proceso de cuatro etapas entre el browser y el servidor; esas etapas son conexión, solicitud, respuesta y desconexión.

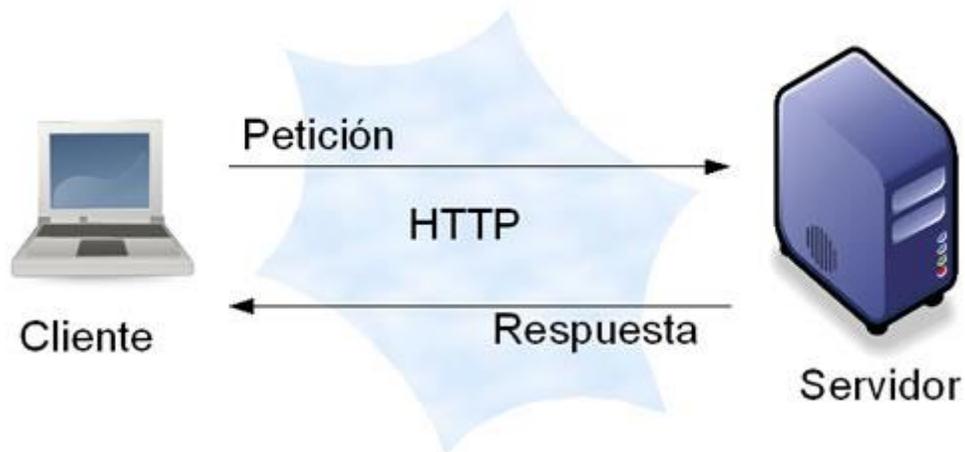


Figura 3. HTTP.

- **Conexión:** el navegador busca el nombre del dominio o el número IP de la dirección indicada intentando hacer contacto con esa computadora.
- **Solicitud:** el navegador envía una petición al servidor, incluyendo información sobre el método a utilizar, la versión del protocolo y algunas otras especificaciones.
- **Respuesta:** el servidor envía un mensaje de respuesta acerca de su petición mediante códigos de estado de tres dígitos.
- **Desconexión:** se puede iniciar por parte del usuario o por parte del servidor una vez transferido un archivo o de haber terminado el proceso de la petición.

2.2.1 Peticiones GET y POST.

El uso adecuado de las peticiones GET y POST es muy importante para que no nos denieguen el acceso a la información, GET es obtener información. Es decir, traer datos que están en el servidor, ya sea en un archivo o base de datos, al cliente. Mientras que el método

POST en enviar información desde el cliente para que sea procesada y actualice o agregue información en el servidor, como sería la carga o actualización en si de una noticia. Cuando enviamos datos a través de un formulario, estos son procesados y luego a través de una redirección por ejemplo devolvemos alguna página de información. (Ardissone, 2011)

2.3 scripts.

Para el proyecto fue necesario la creación de scripts los cuales se encargarían de hacer las descargas y demás procesos. De acuerdo a Figuero (2017) un script es un “fichero de texto con las instrucciones que debe ejecutar el procesador secuencialmente”. Actualmente existen distintos lenguajes de programación capaces de generar scripts, algunos con funciones más potentes que otros, entre los que se considera potentes y que va teniendo auge en el mercado es el lenguaje de Python el cual actualmente se puede encontrar con su versión 3.

2.3.1 Python

Como se mencionó antes Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, está preparado para realizar cualquier otro tipo de programa, desde aplicaciones Windows a servidores de red o incluso páginas web.

Entre sus características esta que es multiplataforma, interpretado, interactivo, orientado a objetos y con una gran mayoría de funciones y librerías (Alvarez, 2003).

2.3.1.1 Web Scraping.

Web scraping es más que nada una técnica que sirve para extraer información de páginas web de forma automatizada. Su uso es muy práctico puesto que sin teclear una sola palabra se puede obtener mucha información en cuestión de segundos. A través de algoritmos de búsqueda

podemos rastrear la información que se necesita. Para poder acceder a los datos de la NASA fue necesario primero crear algoritmos capaces de generar posibles nombres de archivos que se encuentran en sus servidores, es ahí donde entra el web scraping para sacar su información y hacer posible la finalización del proceso. (Martí, 2016)

2.4 Diagrama de Gantt.

Actualmente en los proyectos es necesario tener un control de las actividades a realizar, al tener un control los tiempos de finalización son mejores. Una de las herramientas más utilizadas es las Grafica de Gantt o Diagrama de Gantt la cual fue diseñada por Henry Laurence Gantt a inicios del siglo XIX, esta herramienta nos permite planificar y programar tareas a lo largo de un periodo determinado. Permite realizar el seguimiento y control del progreso de cada una de las etapas del proyecto.

Según Hinojosa (2003) “un diagrama de Gantt consiste en una representación gráfica sobre dos ejes; en el vertical se disponen las tareas del proyecto y en el horizontal se representan el tiempo”.

Entre las características que la distinguen esta que cada actividad se representa mediante un bloque rectangular cuya longitud indica su duración, la altura carece de significado, la posición de cada bloque en el diagrama indica los instantes de inicio y finalización de las tareas a que corresponden, los bloques correspondientes a tareas del camino crítico acostumbran a rellenarse en otro color.

Obviamente como todo tiene sus ventajas y desventajas, su principal ventaja radica en que su trazado requiere un nivel mínimo de planificación, es decir, es necesario que haya un plan que ha de representarse en forma de grafico más, sin embargo, aunque son muy eficaces en las etapas iniciales de la planificación, después de iniciada la ejecución de la actividad y cuando comienza a efectuarse modificaciones, el grafico tiende a volverse confuso.

Actividades	Inicio	Fin	Duración	Enero				Febrero				
				Semana 1	Semana 2	Semana 3	Semana 4	Semana 1	Semana 2	Semana 3	Semana 4	
Estudio de mercado previo	01-01-2017	08-01-2017	1 semana	■								
Constitución del negocio	09-01-2017	23-01-2017	2 semanas		■	■						
Desarrollo de la inversión	24-01-2017	21-02-2017	4 semanas				■	■	■	■		
Construcción del lugar	07-02-2017	21-03-2017	6 semanas						■	■	■	
Amoblado del local	22-03-2017	05-04-2017	2 semanas									
Selección del capital humano	06-04-2017	20-04-2017	2 semanas									
Capacitación del capital humano	20-04-2017	04-05-2017	2 semanas									
Alineación administrativa	05-05-2017	26-05-2017	3 semanas									
Inicio de actividades propias del local	27-05-2017	03-06-2017	1 semana									

Figura 4. Ejemplo de Grafica de Gantt.

2.5 Scrum

La creación de un buen software y su entrega oportuna requieren de un esquema de implementación sistemático mas no excesivamente rígido. Además, se tiene que tener en cuenta tanto la calidad como el tiempo de entrega. Hoy en día dentro de la metodología de desarrollo de software ágil se encuentra el marco de referencia de scrum el cual funciona para proyectos complejos.

En scrum se maneja el concepto de equipos scrum, los cuales son grupos de trabajo donde los miembros juegan roles específicos, en el caso de scrum considera que los Developers son seres humanos que cometen errores, que piensan en nuevas ideas en el camino y muchas características más.

scrum propone realizar el trabajo en cortos ciclos iterativos que van desde una semana hasta un mes, generalmente llamado iteración o sprint. El resultado de cada iteración deberá ser un producto listo para la entrega.

2.6 Técnicas de Recolección de datos

La obtención de información es uno de los pasos más cruciales al iniciar un proyecto. Si desde el principio esta fase no se implementa de la manera apropiada, se tendrán retrasos a futuro por posibles cambios. Entre las técnicas para obtener información se encuentran la encuesta, observación, entrevista, etc.

Según Zapata (2005) la encuesta puede definirse como “un conjunto de técnicas destinadas a reunir, de manera sistemática, datos sobre determinado tema o temas relativos a una población, a través de contactos directos o indirectos con los individuos o grupo de individuos que integran la población estudiada”. Una vez analizado este concepto se puede predecir que es funcional cuando se quiere lanzar un producto nuevo o aplicación y se quiere saber si tendrá impacto en el público al que se quiere dirigir.

Capítulo 3. Metodología

La metodología que se decidió emplear en este proyecto fue la de SCRUM, la cual es una metodología ágil y flexible para gestionar el desarrollo de software. Permite el trabajo colaborativo para obtener los mejores resultados posibles. Por eso mismo se decidió utilizar esta metodología para obtener resultados en periodos cortos.

3.1 Planeación

3.1.1 Gráfica de Gantt.

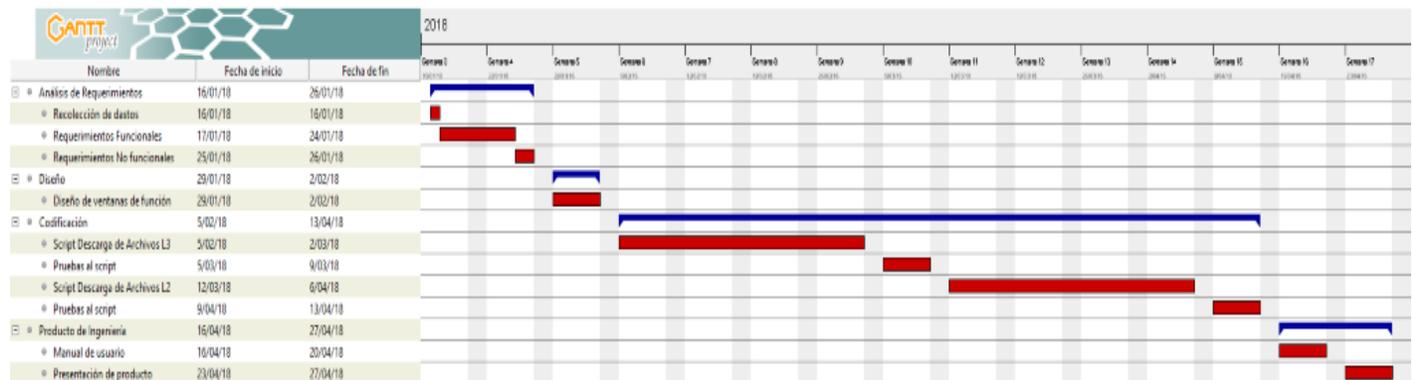


Figura 5. Grafica de Gantt.

Cadena de Descarga Automatica de Datos Satelitales		
12-mar-2018		
2		
Tarea		
Nombre	Fecha de inicio	Fecha de fin
Analisis de Requerimientos	16/01/18	26/01/18
Recolección de datos	16/01/18	16/01/18
Requerimientos Funcionales	17/01/18	24/01/18
Requerimientos No funcionales	25/01/18	26/01/18
Diseño	29/01/18	2/02/18
Diseño de la arquitectura de descarga	29/01/18	2/02/18
Codificación	5/02/18	13/04/18
Script Descarga de Archivos L3	5/02/18	2/03/18
Pruebas al script	5/03/18	9/03/18
Script Descarga de Archivos L2	12/03/18	6/04/18
Pruebas al script	9/04/18	13/04/18
Productos de Ingeniería	16/04/18	27/04/18
Manual de usuario	16/04/18	20/04/18
Presentación de producto	23/04/18	27/04/18

Figura 6. Agenda.

Una gráfica de Gantt es una herramienta que ayuda a planificar y programar tareas a lo largo de un periodo determinado. De igual forma ayuda con el control del recurso que se tiene para llevar a cabo las tareas. Entre las ventajas de usar esta herramienta en la planeación de un proyecto están las siguientes:

- Contribuye a establecer plazos realistas
- Muestra una imagen simple de un sistema complejo
- Ayuda a establecer prioridades
- Es fácil de utilizar y entender

La grafica de Gantt de la figura 1 muestra las actividades y tiempo de duración de cada una con el cual se trabajará en este proyecto. Así mismo la figura 2 a manera de una lista nos muestra las actividades que conlleva el desarrollo del proyecto mostrando los intervalos de duración para cada.

3.1.2 Metodología de desarrollo de software.

Desarrollar software es un proceso el cual necesita un control en su desarrollo, para tener ese control existen diferentes metodologías que están conformadas de una serie de pasos ordenados por orden de prioridad que ayudan a estructurar, planear y controlar el proceso de desarrollo de sistemas informáticos.

Un ejemplo de metodología para el desarrollo de software es el método SCRUM el cual ha sido implementado en el desarrollo de este proyecto, esto es porque la característica principal es que se trabaja por medio de iteraciones y pequeños avances que son presentados cada semana al usuario final, de esta puede dar su opinión acerca de lo que se presenta y si es necesario realizar modificaciones los desarrolladores sabrán que es necesario cambiar.

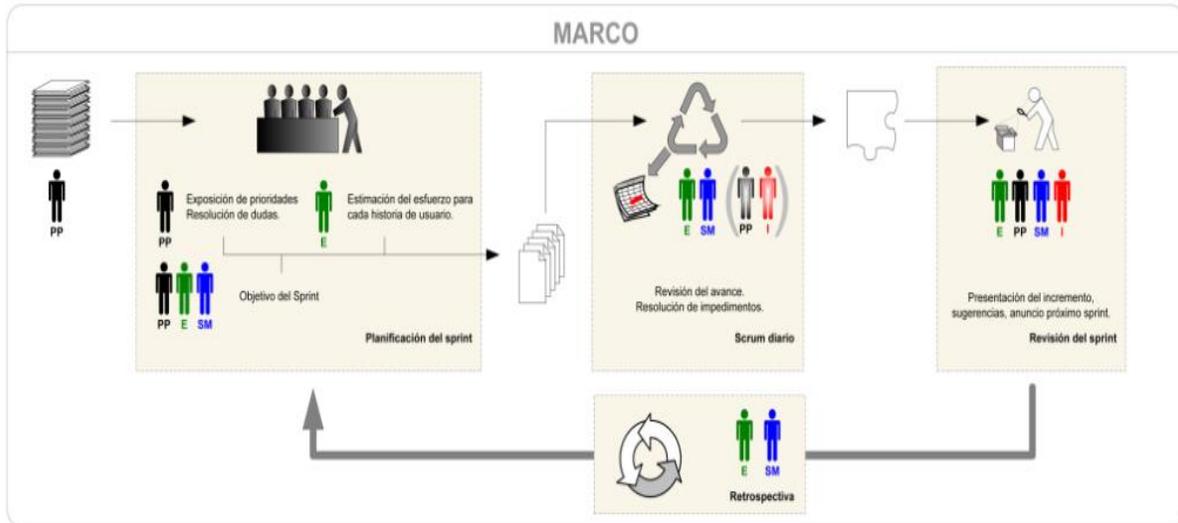


Figura 7. Metodología SCRUM.

3.2 Fase de Análisis

3.2.1 Técnica de recolección de datos.

1. ¿Cuál es la necesidad a satisfacer con el proyecto?

Con el desarrollo de este proyecto se debe permitir a los investigadores del CENIT hacer descargas de archivos de la plataforma de OCEAN COLOR de distintos sensores, estudios, pero todo de una forma automática, de esta forma se tendrá el recurso necesario para las tareas de investigación llevadas a cabo en CENIT.

2. ¿Cuáles son los sensores de los que se descargan los archivos?

Sensor MODIS (Aqua y Terra) y Sensor VIIRS

3. ¿Sobre qué tipo de estudios se realizarán las descargas?

La cadena de descarga debe ser flexible y permitir descargar archivos de cualquier tipo de estudios, aunque la prioridad en este momento es concentración de clorofila chlor_a

4. ¿Sobre qué tipo de nivel de archivos se realizarán las descargas?

Por el momento la descarga de archivos solo será para productos L3 Mapped y Productos L2

5. ¿De qué tipo de resolución se llevarán a cabo las descargas?

Solo de dos tipos de resolución, 4 km por pixel y 9 km por pixel

6. ¿Cómo será la interacción entre el usuario y la cadena de descarga?

Todas las peticiones y solicitudes de descargas serán hechas por medio de consola

7. ¿Los archivos que se descarguen serán guardados en una base de datos?

No, el almacenamiento de los archivos será en un servidor, así que no es necesaria la implementación de un sistema gestor de base de datos, los archivos se deben guardar en una estructura de carpetas las cuales deben estar ordenadas y clasificadas de acuerdo a los tipos de archivos que son descargados.

8. ¿Existirá una hora de descarga específica o se llevarán a cabo cuando el investigador necesite un archivo?

Cuando exista un archivo nuevo en la plataforma de OCEAN COLOR la cadena de procesos debe hacer la descarga en automático, pero a la vez debe permitir al investigador realizar descargas de archivos específicos y necesarios para él.

9. ¿La ruta de almacenamiento siempre será la misma?

No, se debe permitir modificar la ruta para guardar los archivos

10. Mencione algunas especificaciones que debe tener la cadena de proceso de descargas automáticas

- No debe descargar archivos repetidos.
- Debe comprobar que las descargas están hechas correctamente, si hay archivos dañados deben ser remplazados.
- Si ya existe una carpeta creada para cada sensor, los nuevos archivos a descargar deben ser almacenados dentro de esta, al menos que no exista o que la ruta de almacenamiento cambie se debe crear un nuevo directorio.

3.2.2 Requerimientos funcionales y no funcionales.

Requerimientos funcionales

- Descargar archivos de nivel 3 (L3) de la plataforma Ocean Color
- Descargar archivos de nivel 2 (L2) de la plataforma Ocean Color
- Permitir que las descargas sean de archivos específicos (por ej. seleccionar sensor, estudio, rangos de fecha)
- Almacenar los archivos en directorios separados, esto es de acuerdo al tipo de archivo descargado (por ej. año, sensor, estudio, resolución, etc.)
- Permitir al usuario la selección libre del directorio para guardar los archivos que descarga (es importante mencionar que al descargar un paquete de archivos la cadena de descargas crea una carpeta especial para guardarlos asignándole una ubicación por default)

Requerimientos no funcionales

- Contar con un servidor para el almacenamiento de los archivos debido a que serán grandes paquetes descargados y un equipo de este tipo es el más adecuado para el correcto funcionamiento de la cadena y manipulación de archivos
- En el servidor deben estar instalados Python y sus librerías para poder ejecutar la cadena de descarga.

3.3 Fase de Diseño

Descarga de L2

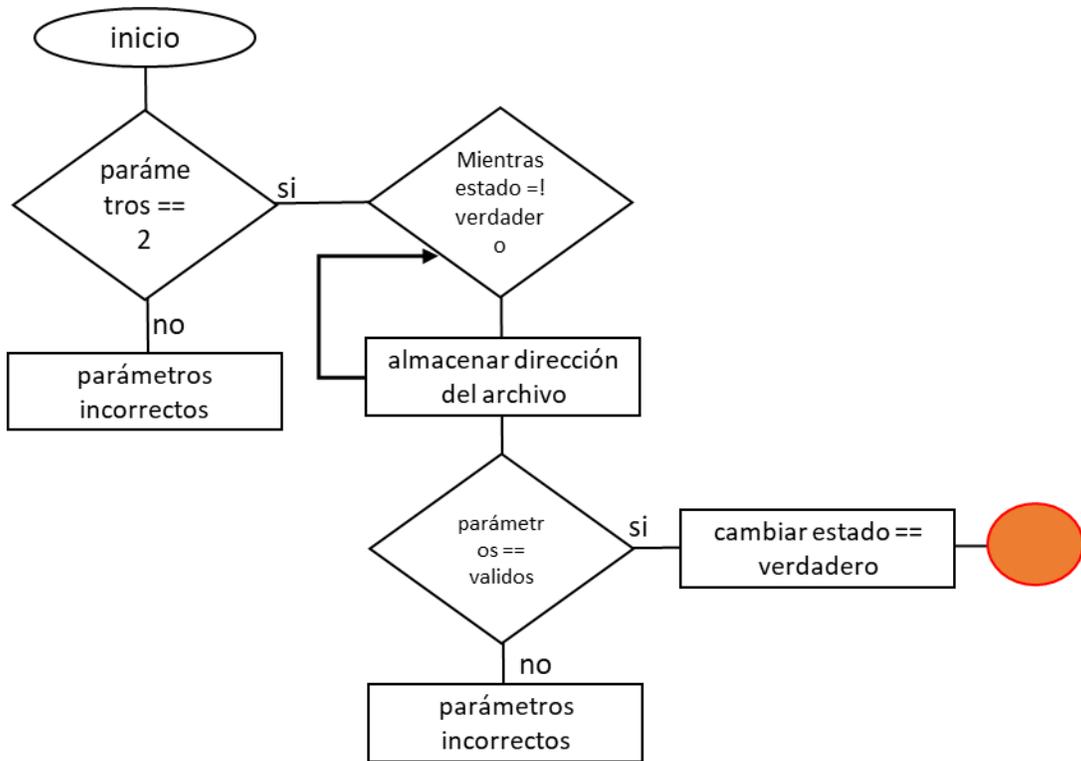


Figura 8. Ejecución Principal de Descarga L2.

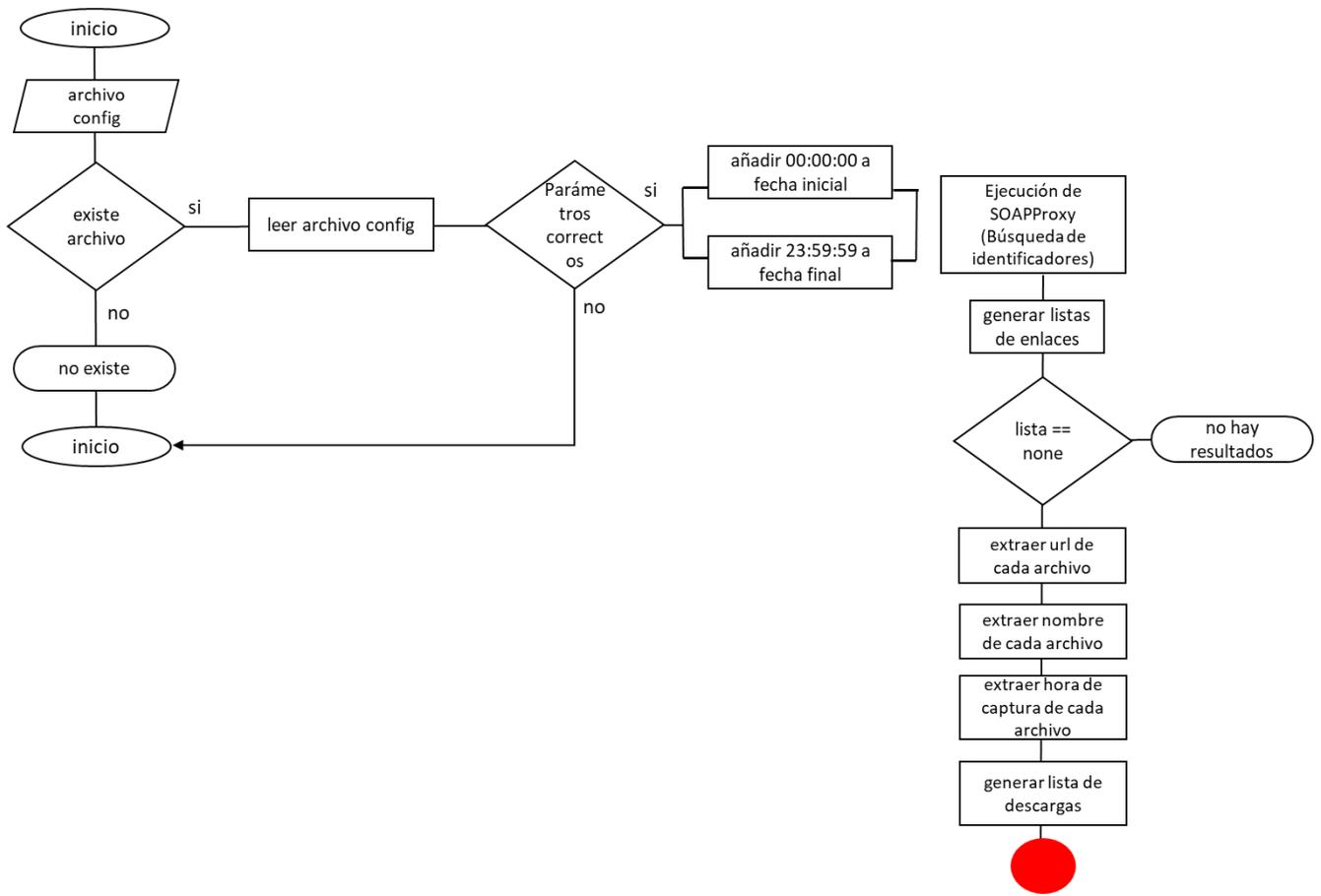


Figura 9. Función Main de Descarga L2.

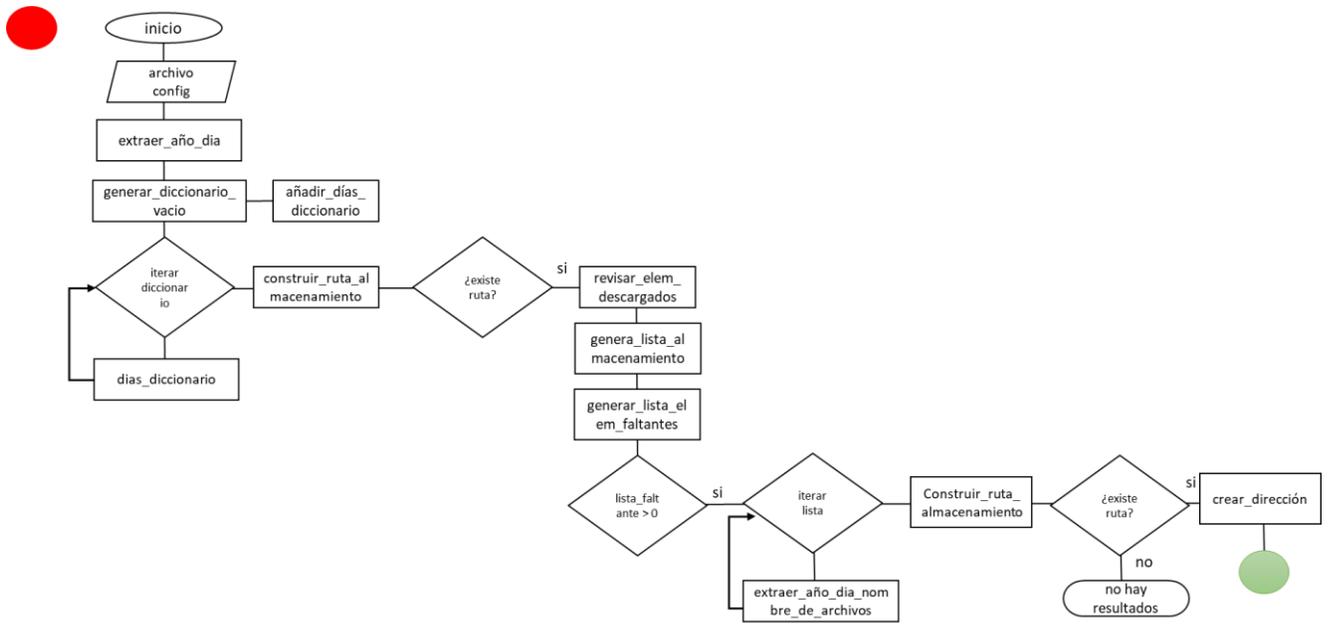


Figura 10. Función encargada de descargar de L2.

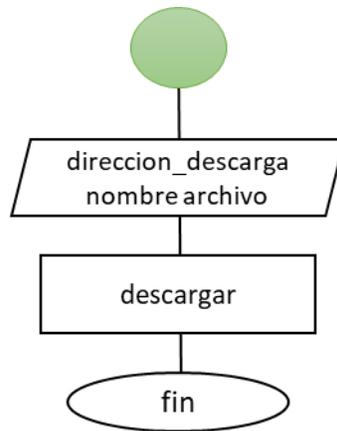


Figura 11. Función Pro_Descargas de descargas L2.

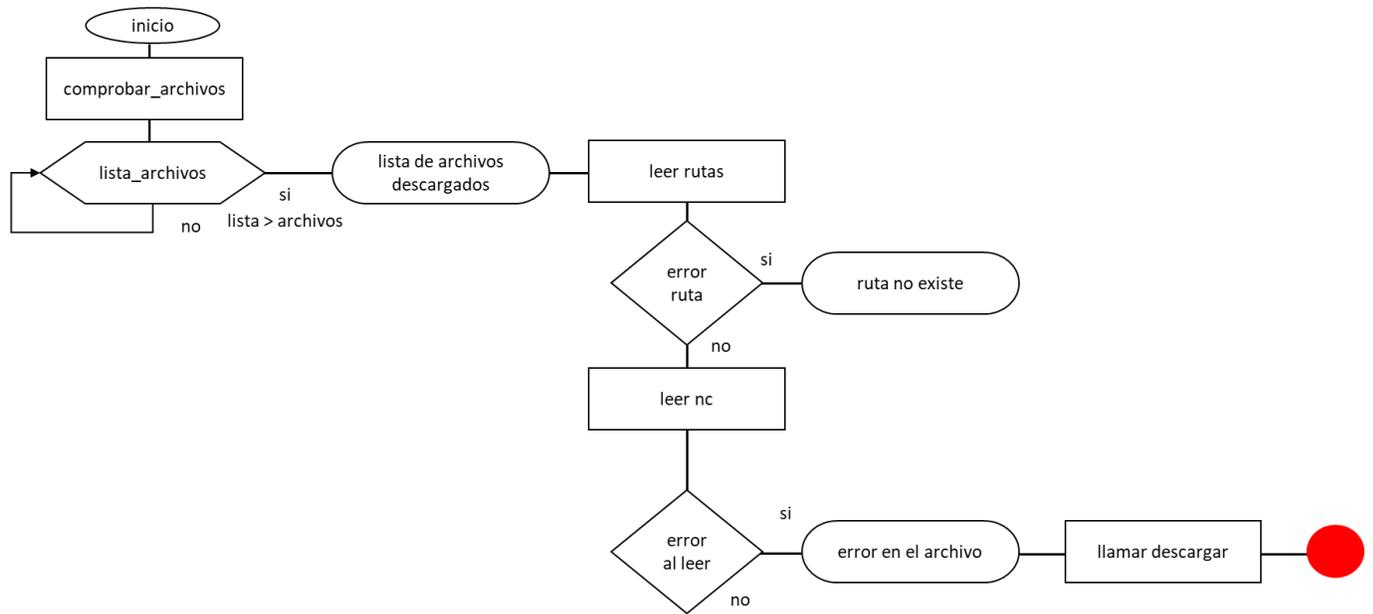


Figura 12. Función para leer los NC de descargas L2.

3.4 Construcción de ingeniería

3.4.1 Codificación.

La cadena de procesamiento es la última fase del proyecto, esta comienza al tener todos los archivos “NC” descargados previamente con la cadena de descarga automatizada de archivos L2.

En esta parte se muestra la función principal del script de procesamiento. Recibe la información necesaria del archivo de configuración. Entre lo que recibe está la ruta de donde se encuentran los NC a procesar. El código saca una lista de esos archivos.

```
1 # -*- coding: utf-8 -*-
2 import netCDF4 as nc
3 import numpy as np
4 import os, sys, logging, makeXml as mx
5
6
7 def hito_processing1km(entrada,rango,output,estudios = None):
8
9     tmp = os.listdir(entrada)
10    newtmp = []
11    for name in tmp:
12        newtmp.append(entrada+"\""+name)
13
14    newtmp2 = []
15    for d in newtmp:
16        result = d.find('.nc')
17        if result != -1:
18            newtmp2.append(d)
19
20    if newtmp2 is None:
21        print "Please select a correct path"
22        return
23    else:
24        pass
25        lista = main(newtmp2,entrada,estudios)
26        rutaXmlFile = mx.makeXML(entrada,lista,rango)
27        processing(newtmp2,entrada,output,rutaXmlFile)
28
```

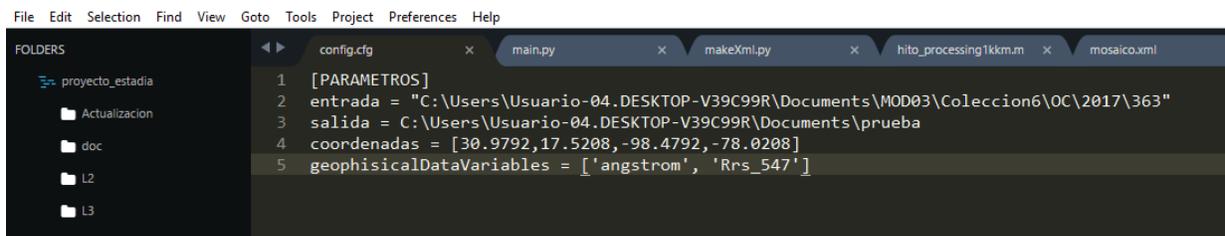
Figura 13. Función Principal de descarga L2.

La función main se encarga de extraer información que se encuentra dentro de los archivos NC. Los grupos que interesan de los archivos son el de geophysical_data y navigation_data.

```
28
29
30
31
32 def main(newtmp2, rutaFichero, estudios = None):
33
34     infoNC = newtmp2[0]
35     dataset = nc.Dataset(infoNC, 'r', format="NetCDF4-BEAM")
36     grupos = dataset.groups.keys()
37
38     g = dataset.groups['geophysical_data']
39     n = dataset.groups['navigation_data']
40
41     varnames = sorted(g.variables.keys())
42
43
44     lista1 = []
45
46     if estudios != None:
47
48         for var in varnames:
49             if var in estudios:
50                 lista1.append(var)
51     else:
52         for var in varnames:
53             lista1.append(var)
54
55     varnames = sorted(n.variables.keys())
56
57     vector2 = ['longitude', 'latitude']
58
59
60
61     for var in varnames:
62         if var in vector2:
63             lista1.append(var)
64
65     return lista1
66
```

Figura 14. Función main de descarga L2.

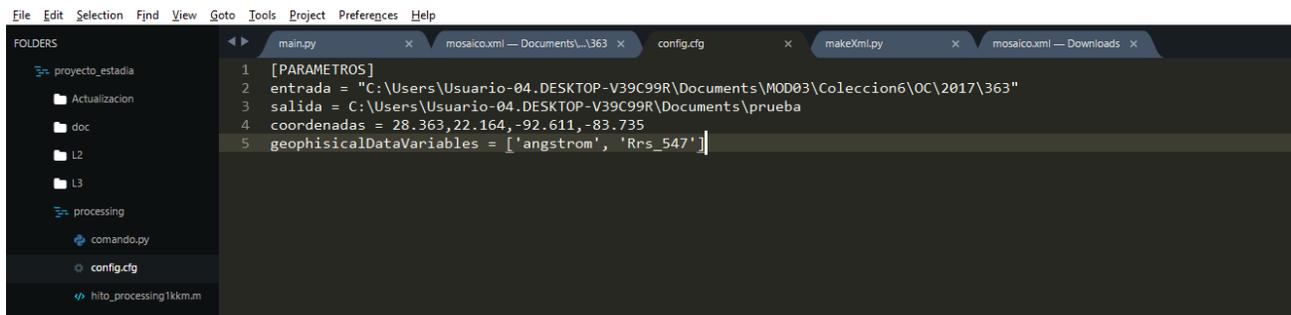
Este es el archivo de configuración donde guarda los diferentes parámetros que se necesitarán para el procesamiento. Entre los datos importantes que se requieren están la ruta donde se encuentran los NC a procesar, la ruta de salida donde se guardará la imagen procesada, las coordenadas que abarcará la imagen y un último parámetro donde se definirá que variables se requieren del grupo de `geophysical_data`, si no se define ninguna entonces por default tomará todas las que contenga este grupo.



```
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
  proyecto_estadia
    Actualizacion
    doc
    L2
    L3
  config.cfg
  main.py
  makeXml.py
  hito_processing1kkm.m
  mosaico.xml
1 [PARAMETROS]
2 entrada = "C:\Users\Usuario-04.DESKTOP-V39C99R\Documents\MOD03\Coleccion6\OC\2017\363"
3 salida = C:\Users\Usuario-04.DESKTOP-V39C99R\Documents\prueba
4 coordenadas = [30.9792,17.5208,-98.4792,-78.0208]
5 geophysicalDataVariables = ['angstrom', 'Rrs_547']
```

Figura 15. Archivo de Configuración de descarga L2.

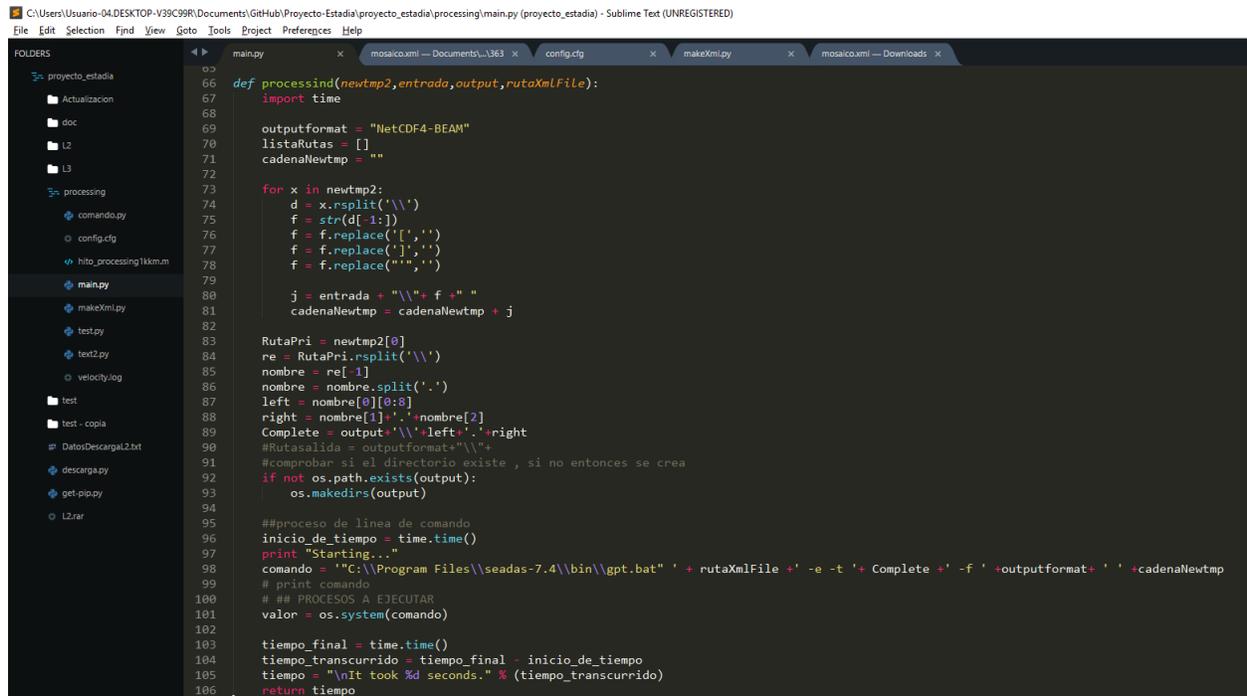
Archivo de configuración para el procesamiento de los datos de nivel 2 (L2), este contiene parámetros que serán necesarios al momento de ejecutar el script, como la ruta de la carpeta donde se encuentran los NC a procesar, la ruta donde se desea guardar la imagen procesada, las coordenadas y opcionalmente si se desea algunas variables específicamente del grupo de geophysicalData se deben especificar.



```
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
  proyecto_estadia
    Actualizacion
    doc
    L2
    L3
  processing
    comando.py
    config.cfg
    hito_processing1kkm.m
  main.py
  mosaico.xml — Documents\...363
  config.cfg
  makeXml.py
  mosaico.xml — Downloads
1 [PARAMETROS]
2 entrada = "C:\Users\Usuario-04.DESKTOP-V39C99R\Documents\MOD03\Coleccion6\OC\2017\363"
3 salida = C:\Users\Usuario-04.DESKTOP-V39C99R\Documents\prueba
4 coordenadas = 28.363,22.164,-92.611,-83.735
5 geophysicalDataVariables = ['angstrom', 'Rrs_547']
```

Figura 16. Archivo de configuración de Procesamiento L2.

Esta función recibe distintos parámetros los cuales se han estado creando en otras funciones. Al recibirlas entonces comienza con la ejecución de un comando en cmd.

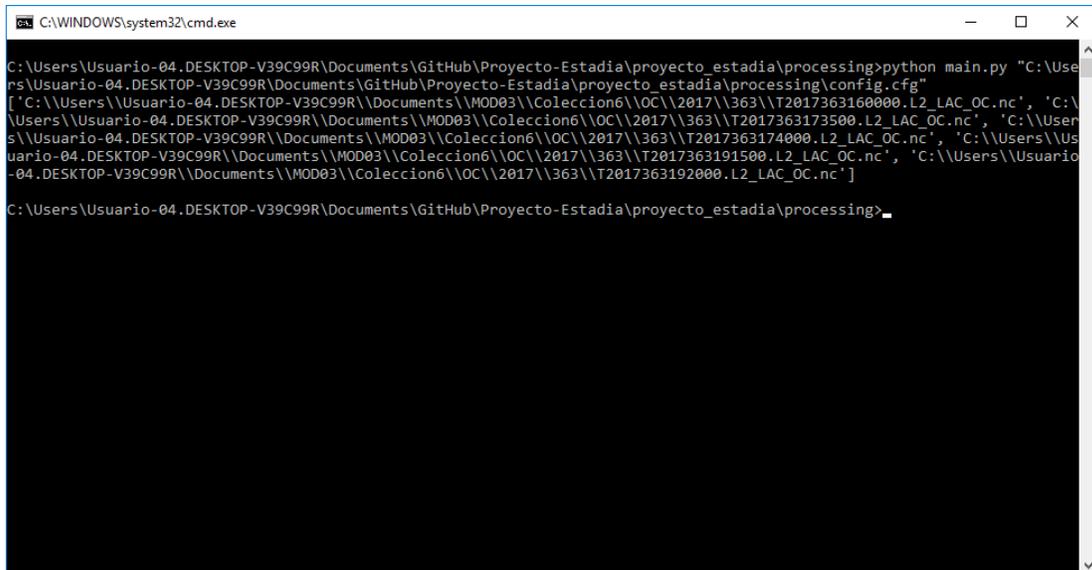


```
66 def processind(newtmp2, entrada, output, rutaXmlFile):
67     import time
68
69     outputformat = "NetCDF4-BEAM"
70     listaRutas = []
71     cadenaNewtmp = ""
72
73     for x in newtmp2:
74         d = x.rsplit('\\')
75         f = str(d[-1])
76         f = f.replace('[', '')
77         f = f.replace(']', '')
78         f = f.replace("'", '')
79
80         j = entrada + "\\\" + f + " "
81         cadenaNewtmp = cadenaNewtmp + j
82
83     RutaPri = newtmp2[0]
84     re = RutaPri.rsplit('\\')
85     nombre = re[-1]
86     nombre = nombre.split('.')
87     left = nombre[0][0:8]
88     right = nombre[1]+'.'+nombre[2]
89     Complete = output+'\\'+left+'.'+right
90     #Rutasalida = outputformat+"\\\"+
91     #comprobar si el directorio existe , si no entonces se crea
92     if not os.path.exists(output):
93         os.makedirs(output)
94
95     ##proceso de linea de comando
96     inicio_de_tiempo = time.time()
97     print "Starting.."
98     comando = "C:\\Program Files\\seadas-7.4\\bin\\gpt.bat" + rutaXmlFile + ' -e -t '+ Complete + ' -f ' +outputformat+ ' ' +cadenaNewtmp
99     # print comando
100     ## PROCESOS A EJECUTAR
101     valor = os.system(comando)
102
103     tiempo_final = time.time()
104     tiempo_transcurrido = tiempo_final - inicio_de_tiempo
105     tiempo = "\nIt took %d seconds." % (tiempo_transcurrido)
106     return tiempo
```

Figura 17. Función que se encarga de procesar.

3.4.2 Pruebas del Script

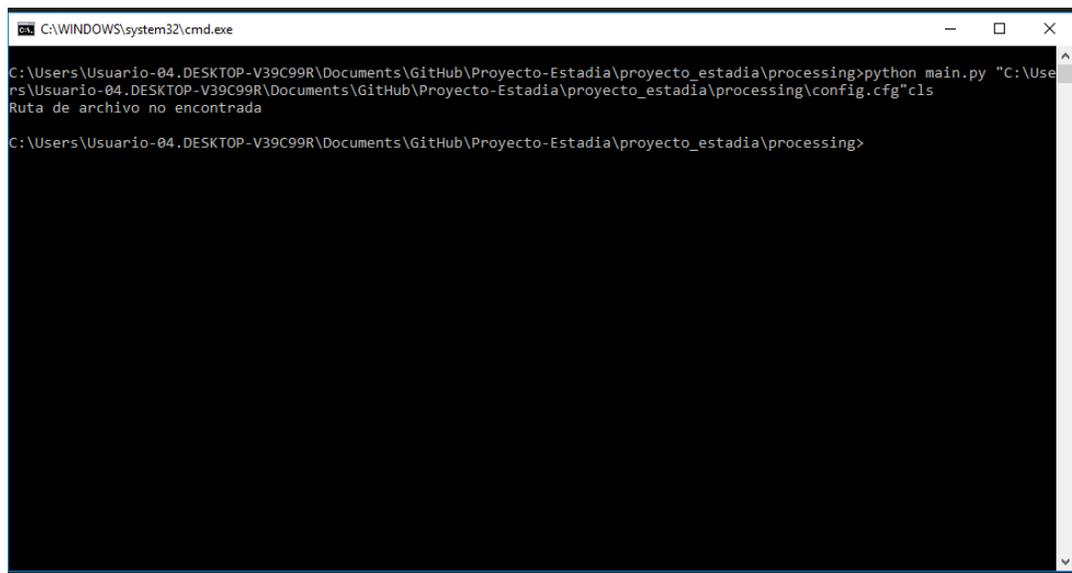
Prueba para saber si la función principal me devuelve la lista de los NC que se encuentran en la carpeta que se especificó en el archivo de configuración.



```
C:\WINDOWS\system32\cmd.exe
C:\Users\Usuario-04.DESKTOP-V39C99R\Documents\GitHub\Proyecto-Estadia\proyecto_estadia\processing>python main.py "C:\Users\Usuario-04.DESKTOP-V39C99R\Documents\GitHub\Proyecto-Estadia\proyecto_estadia\processing\config.cfg"
["C:\\Users\\Usuario-04.DESKTOP-V39C99R\\Documents\\MOD03\\Coleccion6\\OC\\2017\\363\\T2017363160000.L2_LAC_OC.nc", 'C:\\Users\\Usuario-04.DESKTOP-V39C99R\\Documents\\MOD03\\Coleccion6\\OC\\2017\\363\\T2017363173500.L2_LAC_OC.nc', 'C:\\Users\\Usuario-04.DESKTOP-V39C99R\\Documents\\MOD03\\Coleccion6\\OC\\2017\\363\\T2017363174000.L2_LAC_OC.nc', 'C:\\Users\\Usuario-04.DESKTOP-V39C99R\\Documents\\MOD03\\Coleccion6\\OC\\2017\\363\\T2017363191500.L2_LAC_OC.nc', 'C:\\Users\\Usuario-04.DESKTOP-V39C99R\\Documents\\MOD03\\Coleccion6\\OC\\2017\\363\\T2017363192000.L2_LAC_OC.nc']
C:\Users\Usuario-04.DESKTOP-V39C99R\Documents\GitHub\Proyecto-Estadia\proyecto_estadia\processing>
```

Figura 18. Lista de los NC.

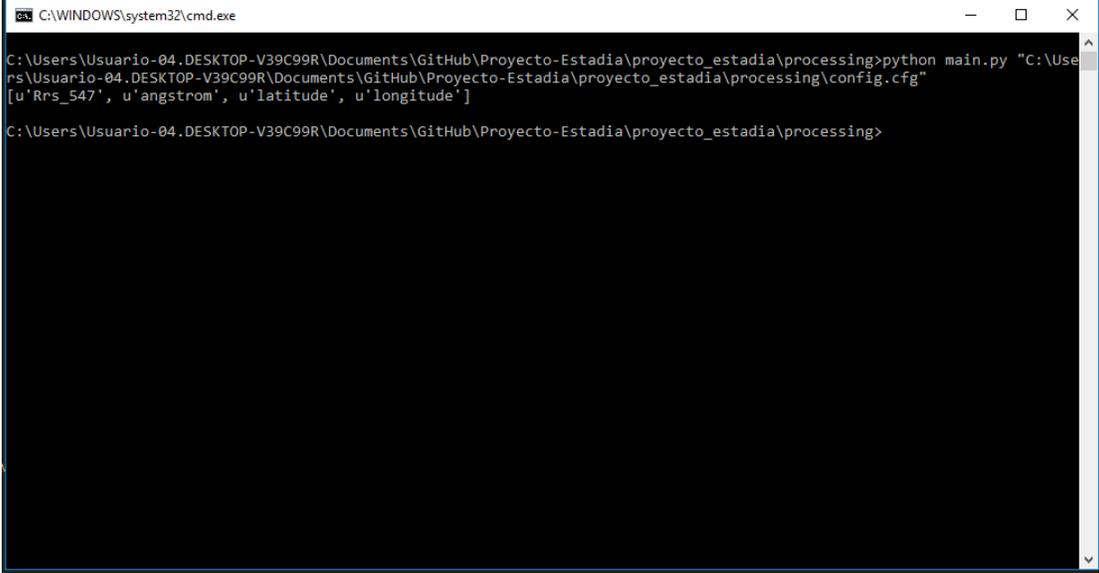
Validación funcionando en caso de introducir erróneamente la ruta del archivo de configuración. Envía un mensaje diciendo “Ruta de archivo no encontrada”.



```
C:\WINDOWS\system32\cmd.exe
C:\Users\Usuario-04.DESKTOP-V39C99R\Documents\GitHub\Proyecto-Estadia\proyecto_estadia\processing>python main.py "C:\Users\Usuario-04.DESKTOP-V39C99R\Documents\GitHub\Proyecto-Estadia\proyecto_estadia\processing\config.cfg"cls
Ruta de archivo no encontrada
C:\Users\Usuario-04.DESKTOP-V39C99R\Documents\GitHub\Proyecto-Estadia\proyecto_estadia\processing>
```

Figura 19. Error de Ruta de Archivo no encontrada.

Comprobación de que la función main funcione correctamente y devuelva una lista con las variables extraídas de los NC.



```
C:\WINDOWS\system32\cmd.exe
C:\Users\Usuario-04.DESKTOP-V39C99R\Documents\GitHub\Proyecto-Estadia\proyecto_estadia\processing>python main.py "C:\Users\Usuario-04.DESKTOP-V39C99R\Documents\GitHub\Proyecto-Estadia\proyecto_estadia\processing\config.cfg"
[u'Rrs_547', u'angstrom', u'latitude', u'longitude']
C:\Users\Usuario-04.DESKTOP-V39C99R\Documents\GitHub\Proyecto-Estadia\proyecto_estadia\processing>
```

Figura 20. Ejemplo de Variables Extraídas de los NC.

Comprobación de la función makeXML, esta se encarga de crear un XML basándose en los datos introducidos en el archivo de configuración como coordenadas y en las variables extraídas de los NC en funciones anteriores. Además, devuelve la ruta donde se guardó el XML.

```
Selecionar C:\WINDOWS\system32\cmd.exe
C:\Users\Usuario-04.DESKTOP-V39C99R\Documents\GitHub\Proyecto-Estadia\proyecto_estadia\processing>python main.py "C:\Users\Usuario-04.DESKTOP-V39C99R\Documents\GitHub\Proyecto-Estadia\proyecto_estadia\processing\config.cfg"
C:\Users\Usuario-04.DESKTOP-V39C99R\Documents\MOD03\Coleccion6\OC\2017\363\mosaico.xml
C:\Users\Usuario-04.DESKTOP-V39C99R\Documents\GitHub\Proyecto-Estadia\proyecto_estadia\processing>
```

Figura 21. Comprobación para saber si devolvió la ruta del XML.

```
config.cfg x main.py x makeXml.py x hto_processing\kktm x mosaico.xml
1 <?xml version="1.0" encoding="utf-8"?>
2 <graph id="PERSEO">
3 <version>1.0</version>
4 <node id="Nodo1">
5 <operator>Mosaic</operator>
6 <sources>
7 <sourceProducts>${sourceProducts}</sourceProducts>
8 </sources>
9 <parameters>
10 <variables>
11 <variable>
12 <name>chlon_a</name>
13 <expression>chlon_a</expression>
14 </variable>
15 <variable>
16 <name>longitude</name>
17 <expression>longitude</expression>
18 </variable>
19 <variable>
20 <name>latitude</name>
21 <expression>latitude</expression>
22 </variable>
23 </variables>
24 <combine>OR</combine>
25 <crs>GEOGCS["WGS84(DD)",DATUM["WGS84",SPHEROID["WGS84",6378137.0,298.257223563]],PRIMEM["Greenwich",0.0],UNIT["degree",0.017453292519943295],AXIS["Geodetic longitude",EAST],AXIS["Geodetic latitude",NORTH]]</crs>
26 <orthorectify>false</orthorectify>
27 <elevationModelName>GETASSE30</elevationModelName>
28 <resampling>Nearest</resampling>
29 <westBound>-98.48</westBound>
30 <northBound>30.98</northBound>
31 <eastBound>-78.02</eastBound>
32 <southBound>17.52</southBound>
33 <pixelSizeX>0.01</pixelSizeX>
34 <pixelSizeY>0.01</pixelSizeY>
35 </parameters>
36 </node>
37 </graph>
```

Figura 22. Ejemplo de XML Creado.

Comprobación que el procesamiento termine de forma exitosa.

```
Seleccionar C:\WINDOWS\system32\cmd.exe
C:\Users\Usuario-04.DESKTOP-V39C99R\Documents\GitHub\Proyecto-Estadia\proyecto_estadia\processing>python main.py "C:\Users\Usuario-04.DESKTOP-V39C99R\Documents\GitHub\Proyecto-Estadia\proyecto_estadia\processing\config.cfg"
Starting...

It took 28 seconds.

C:\Users\Usuario-04.DESKTOP-V39C99R\Documents\GitHub\Proyecto-Estadia\proyecto_estadia\processing>
```

Figura 23. Prueba en consola de Procesamiento terminada.

Archivo que se crea al momento de procesar las imágenes.

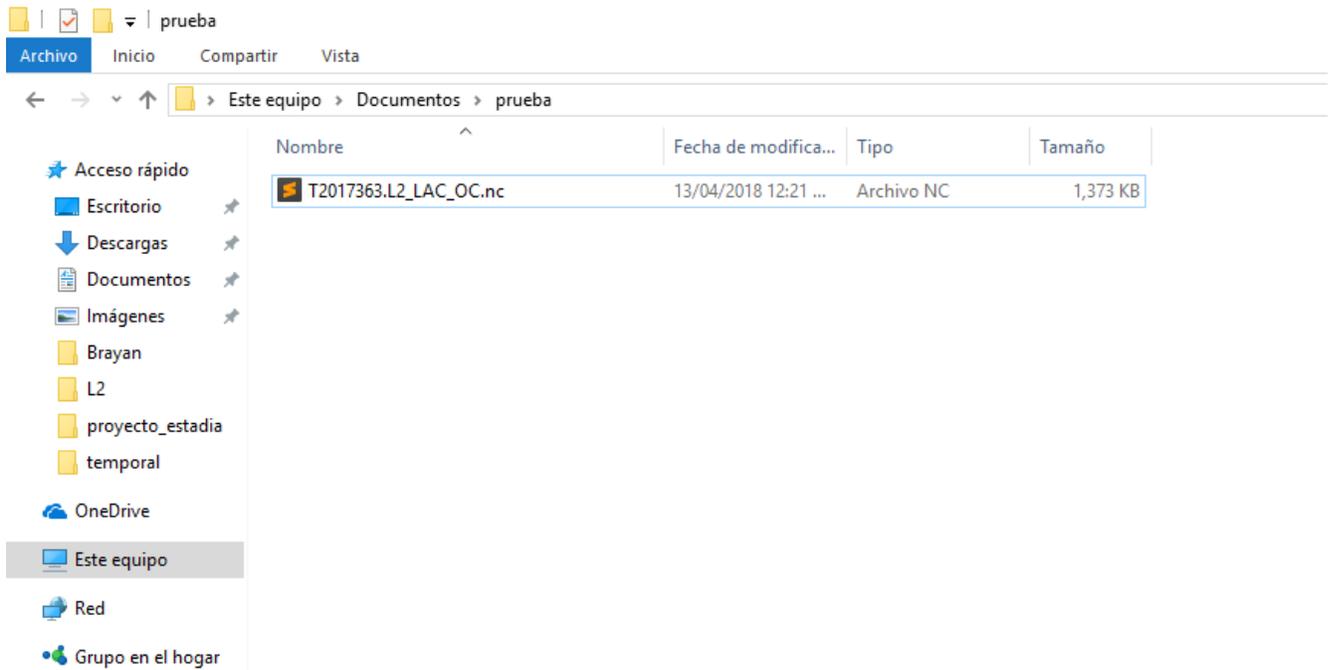
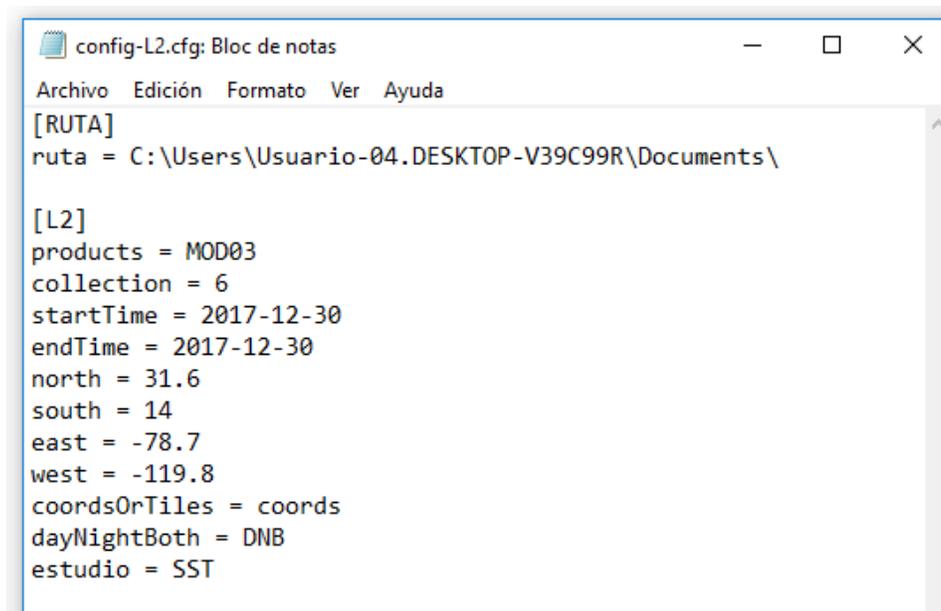


Figura 24. Archivo Procesado.

Capítulo 4. Resultados

Parámetros del archivo de configuración para la descarga de archivos de la NASA de nivel 2 (L2). De acuerdo a lo introducido se está especificando que se desea descargar productos del sensor MODIS Terra de la colección 6 del día 30 de diciembre del 2017 con las coordenadas para norte: 31.6, sur: 14, este: -78.7, oeste: -119.8, de los datos extraídos durante el día y la noche del tipo de estudio de la temperatura de la superficie del mar.



```
config-L2.cfg: Bloc de notas
Archivo Edición Formato Ver Ayuda
[RUTA]
ruta = C:\Users\Usuario-04.DESKTOP-V39C99R\Documents\

[L2]
products = MOD03
collection = 6
startTime = 2017-12-30
endTime = 2017-12-30
north = 31.6
south = 14
east = -78.7
west = -119.8
coordsOrTiles = coords
dayNightBoth = DNB
estudio = SST
```

Figura 25. Archivo de configuración para L2.

Aquí se puede apreciar como la cadena de descarga de archivos de la NASA de nivel 2 (L2) se termina de ejecutar, descargando los 6 NC disponibles de ese día que se especificó.

```
C:\WINDOWS\system32\cmd.exe
C:\Users\Usuario-04_DESKTOP-V39C99R\Documents\GitHub\Proyecto-Estadia\proyecto_estadia\L2>python main.py "C:\Users\Usuario-04_DESKTOP-V39C99R\Documents\GitHub\Proyecto-Estadia\proyecto_estadia\L2\config-L2.cfg"
Iniciando...
Iniciando descarga
Descargando 6 de 6
C:\Users\Usuario-04_DESKTOP-V39C99R\Documents\GitHub\Proyecto-Estadia\proyecto_estadia\L2>
```

Figura 26. Descargando Archivos.

Muestra de los archivos NC una vez descargados dentro de su carpeta correspondiente.

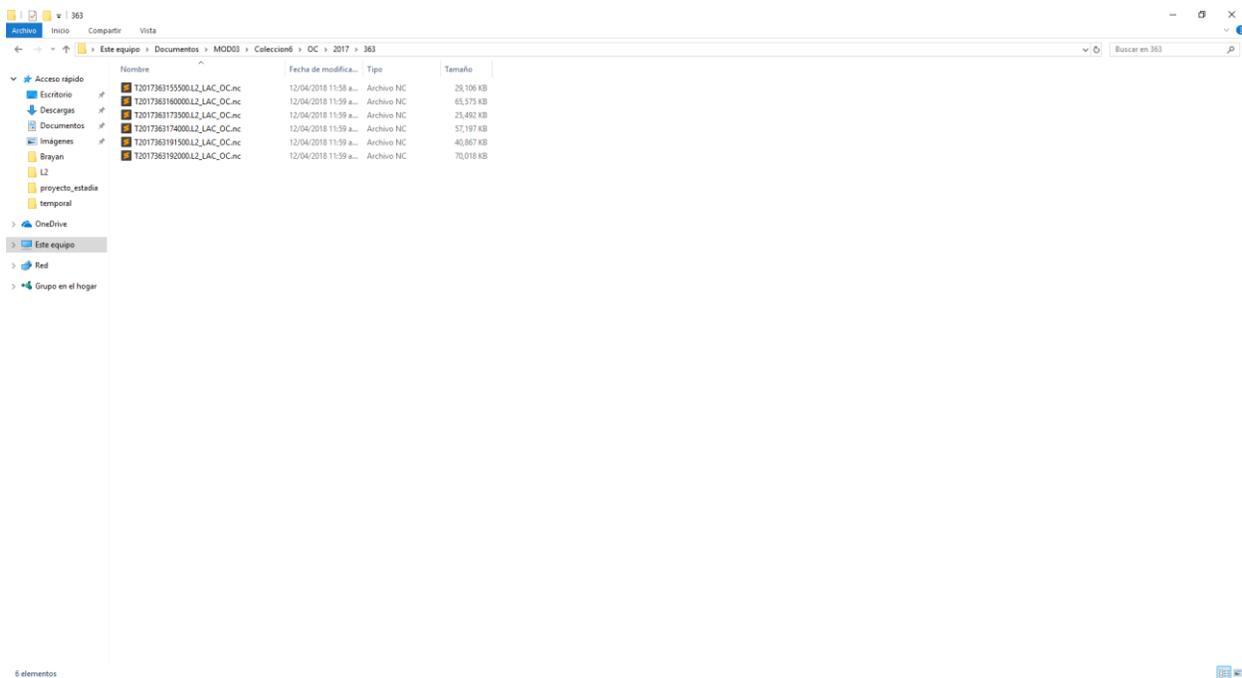


Figura 27. Archivos NC dentro del directorio.

Archivos NC L2 del mismo día descargados, antes de ser procesados.

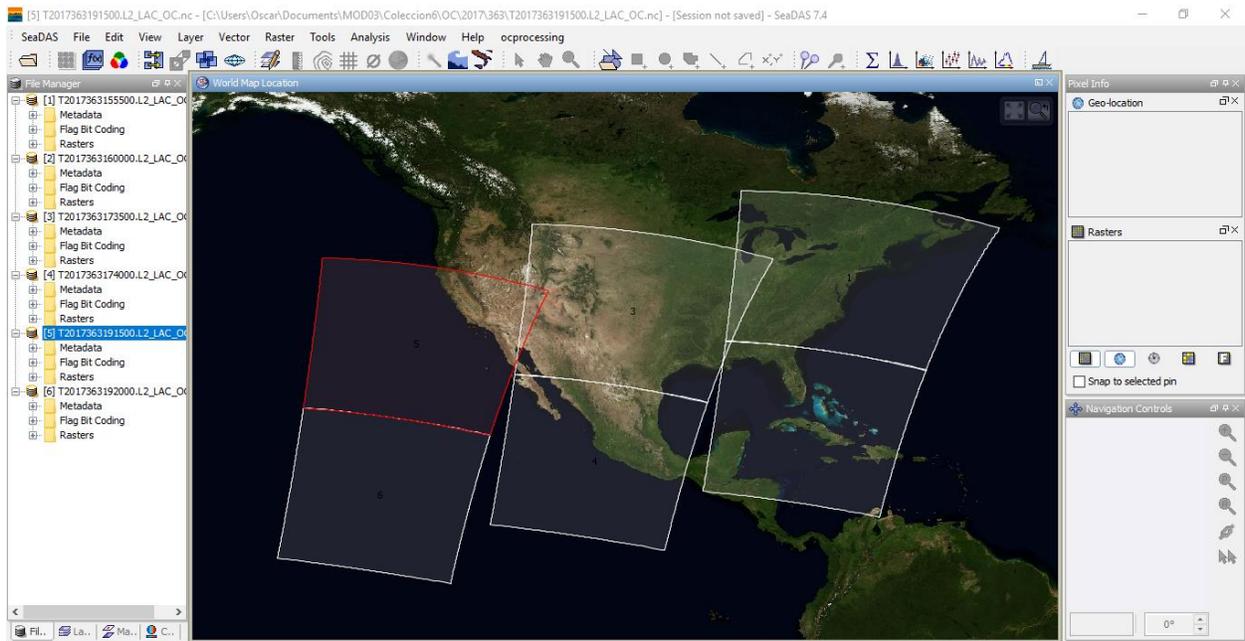


Figura 28. Imágenes L2 por procesar.

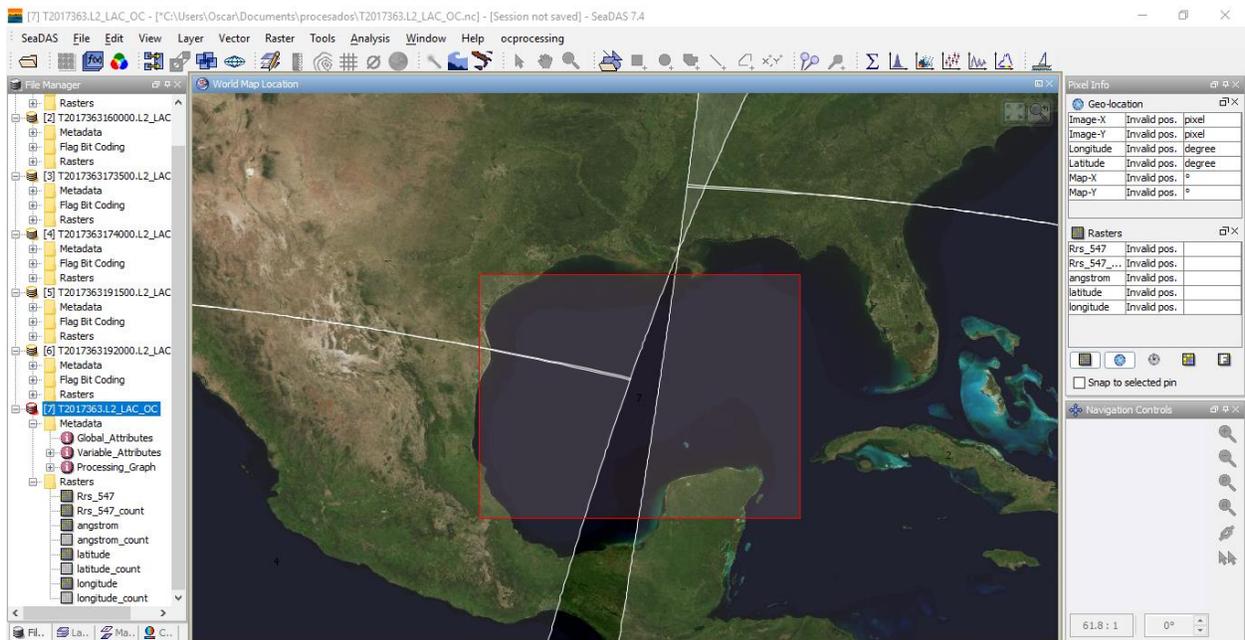


Figura 29. NC procesado.

En esta podemos observar los distintos datos que se especificaron al descargar el archivo NC.

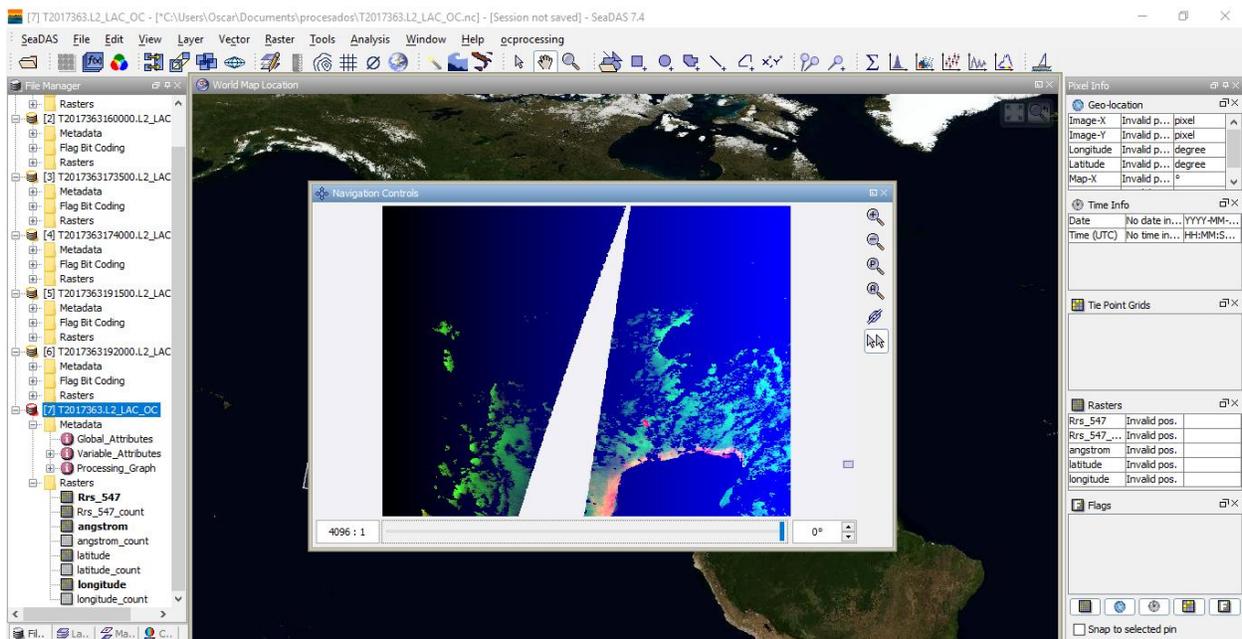


Figura 30. Muestra de un NC de L2 convertido a L3.

Capítulo 5. Conclusiones

En general el proceso de estadías se realizó con éxito ya que se cumplieron los objetivos planteados al inicio del proyecto. Gracias a esto ahora en el área de percepción remota satelital del CENIT tendrá un recurso eficiente para realizar descarga de datos satelitales de forma automática. De igual forma los recursos que sean descargados estarán organizados de una forma ordenada haciendo eficiente su búsqueda cuando exista la necesidad de consultarlos para realizar los trabajos de investigación. Debido a que hubo una buena comunicación con el Dr. Juan Pablo Rivera Caicedo investigador del CENIT, se realizó un análisis detallado de los requerimientos que era necesario satisfacer con el desarrollo de este proyecto, gracias a la buena comunicación se aclararon las dudas que surgían mediante el desarrollo del proyecto.

Durante el proceso de estadía se presentaron algunas dificultades técnicas como lo fueron ciertas actualizaciones al OCEAN COLOR las cuales impedían la descarga de archivos y un

detalle con el funcionamiento de los algoritmos de descargas en la plataforma Linux, pero no fueron lo suficientemente fuertes para no concluir con el proyecto. Tras la realización de la cadena de descargas automáticas de datos satelitales de nivel 2 (L2) y nivel 3 (L3) se presentó la cadena de descargas al Dr. Caicedo responsable del proyecto quien evaluó el trabajo realizado. Es importante mencionar que a lo largo del proyecto el Dr. Caicedo estuvo revisando avances y haciendo propuestas de mejora por lo cual se realizaron las modificaciones necesarias para el correcto funcionamiento del algoritmo de descarga.

En cuanto a la forma en la que ahora se podrán descargar los archivos se les hizo bien, esto porque ahora podrán descargar más de un archivo a la vez y con las especificaciones necesarias a lo que desean investigar. Es importante mencionar que el Dr. Caicedo hizo énfasis en que los datos que se descargaran deberían estar ordenados en una serie de directorios creados de acuerdo a las características de los datos descargados, esto es para hacer más fácil el procesado y la búsqueda de los datos que sea necesario utilizar debido a que estarán organizados por los tipos de estudios, por fechas, resolución, etc. Esta cadena ahorra tiempo a los investigadores en realizar sus descargas.

En relación a la configuración de descargas fue necesario crear un archivo de configuración y en él se escriben una serie de parámetros necesarios para especificar lo que se desea descargar o procesar según sea el caso del script que se ejecute. Es importante mencionar que los parámetros cambian de acuerdo al nivel de los archivos que se desean descargar. Este archivo es llamado en la función principal de la cadena de descarga para comenzar a bajar los archivos necesarios.

El desarrollo del proyecto presento algunas limitaciones en cuestión de tiempo, anteriormente se había mencionado al inicio de este documento que el tiempo sería un factor no

favorable para la documentación completa del proyecto, ya que para desarrollarlo completamente se estimó un plazo de seis meses pero el proceso de estadías consta de cuatro sumado a esto está el periodo vacacional que consta de la última semana del mes de marzo y la primera del mes de abril, por lo que se optó por solo documentar los avances que se tengan hasta la fecha de conclusión de estadías establecida en la agenda que facilitó al inicio del proceso, este detalle se comentó con el asesor empresarial y académico los cuales no expresaron alguna objeción estando de acuerdo en que el proyecto se documente hasta la fecha del 27 de abril del año en curso. Pero es importante mencionar que los objetivos establecidos a cuatro meses se cumplieron de forma exitosa los cuales constan de la elaboración de los algoritmos de descarga para los archivos de nivel 2 y nivel 3. Un dato extra es que la cadena de descarga de archivos de nivel 3 se comenzó a ser utilizada por estudiantes de postgrado antes de la fecha que se estimó para ser terminada y en cuanto se terminó de hacer el algoritmo de descarga de nivel 2 también se facilitó a estudiantes e investigadores del CENIT para que pudieran usarlo. Cabe mencionar que este proyecto no solo será utilizado por los investigadores del CENIT como se había planteado en un inicio, ahora se facilitara a todos los estudiantes de postgrado y practicantes de licenciatura o ingeniería que lo necesiten para realizar sus tesis o investigaciones por esto fue que los algoritmos de descargas se realizaron antes del mes de abril.

Como recomendación para el correcto funcionamiento de la cadena de descarga automática de datos satelitales es necesario ser muy cuidadoso a la hora de llenar el apartado de parámetros en el archivo de configuración ya que de los datos que en él se ingresen dependen los archivos que se desean descargar o procesar.

Aún terminado el proyecto ya se tiene planeado realizar más proyectos que ayuden a la investigación en CENIT, algunos relacionados a la programación y otros con la electrónica,

hacer uso de tarjetas Arduino etc. Todos con la misma finalidad, ayudar a los investigadores en su trabajo científico.

Igualmente, el algoritmo de descargas realizado puede ser ampliarse más, por ejemplo, en el procesamiento de todas las imágenes satelitales que sean descargadas ya que por el momento solo se descargan las imágenes de nivel 2 (L2) pero como se mencionó anteriormente la falta de tiempo es un factor que impide que se desarrolle completamente.

Como conclusión final, el desarrollo del proyecto que consta de la creación de algoritmos de descarga fue llevado de forma exitosa cumpliendo con los requerimientos establecidos por el Dr. Caicedo hasta el 27 de abril del 2018.

Referencias

- Alvarez, M. A. (19 de Noviembre de 2003). *desarrolloweb.com*. Obtenido de desarrolloweb.com: <https://desarrolloweb.com/articulos/1325.php>
- Ardissone, J. (9 de Febrero de 2011). *blog.micayael.com*. Obtenido de blog.micayael.com: <http://blog.micayael.com/2011/02/09/metodos-get-vs-post-del-http/>
- Dimes, T. (2015). *Conceptos Basicos De Scrum: Desarrollo De Software Agile Y Manejo De Proyectos Agile*. Babelcube Inc.
- Figuro, C. G. (2017). *El programa R, herramienta clave en investigación*. Santander: Universidad Cantabria.
- Hinojosa, M. A. (11 de Marzo de 2003). *www.gestiopolis.com*. Obtenido de www.gestiopolis.com: <https://www.gestiopolis.com/diagrama-de-gantt/#pf3>
- Martí, M. (8 de Abril de 2016). *sitelabs.es*. Obtenido de sitelabs.es: <https://sitelabs.es/web-scraping-introduccion-y-herramientas/>
- Mas, Jean-Francois. (Septiembre de 2011). *www.ciga.unam.mx*. Obtenido de www.ciga.unam.mx: http://www.ciga.unam.mx/publicaciones/images/abook_file/aplicacionesMODIS.pdf

Murillo Sandoval, P. J., & Carbonell González, J. A. (Mayo de 2012). *Principios y aplicaciones de la percepción remota en el cultivo de la caña de azúcar en Colombia*. Cali: Feriva S.A. Obtenido de cenicana.

Olmo, P. C. (s.f.). <http://www.ujaen.es>. Obtenido de <http://www.ujaen.es>: <http://www.ujaen.es/huesped/pidoceps/telap/perremoc.htm>

www.educoas.org. (s.f.). Obtenido de www.educoas.org: <https://www.educoas.org/portal/bdigital/contenido/valzacchi/ValzacchiCapitulo-2New.pdf>

Zapata, O. A. (2005). *La aventura del pensamiento crítico : herramientas para elaborar tesis e investigaciones socioeducativas*. Mexico: pax.

Anexos

Redacción de procesos de L2

Notas de Simbología

- ↘ Comienzo de proceso
- Subnivel individual
- Llamada de funciones o scripts
- ⇔ Subnivel corrido

Primera parte ↘

Ejecución principal:

- Define estado como falso
- Si la longitud de los parámetros es igual a dos:

- ☐ Mientras estado no sea verdadero:
 - ☐ Almacenar la dirección del archivo en variable
 - ☐ Si la variable es diferente de nada:
 - ☐ Si la variable es un archivo válido:
 - ☐ estado es a verdadero
- ☐ Sino:
 - ☐ Imprimir “Parámetros incorrectos”
- ☐ Si estado es verdadero:
 - ☐ Ejecuta función *main*(dirección del archivo de configuración)

Función *main*

Leer dirección de archivo de configuración

Obtener parámetros y guardarlos en variables

Llamada al script *validar_datos* función *validar*

Si la función retorna verdadero:

⇒ Añadir ‘00:00:00’ a la variable de fecha inicial

Añadir ‘23:59:59’ a la variable de fecha final

Condiciones para determinar el sensor seleccionado

Ejecución de *SOAPProxy*(enlace de página web API MODAPS)

Búsqueda de identificadores mediante el cliente *SOAPProxy*

Generación de lista con los ID’s obtenidos

Si la lista de ID’s es diferente de nada:

☐ Bucle para iterar la lista de ID’s:

⇒ Obtener url del archivo con el ID actual

De cada url extraer el nombre del archivo

De cada nombre de archivo extraer la hora de captura

Condiciones para determinar el estudio seleccionado

Generar lista con los enlaces de descarga de cada archivo encontrado

Llamada al script y función *descarga*

Sino:

Imprime “No hay resultados”

Script *descarga*

Leer dirección de archivo de configuración

Obtener la ruta de almacenamiento dentro de la configuración

Bucle para iterar los enlaces de descarga:

Extraer año y día

Generar diccionario con una lista vacía por defecto

Añadir los días a la lista dentro del diccionario

Bucle para iterar los elementos del diccionario:

Bucle para iterar los días dentro de los elementos del diccionario:

Construir la dirección de almacenamiento

Si la dirección existe:

Revisa los elementos descargados dentro de la carpeta

Genera lista con la dirección de almacenamiento

Se genera lista de archivos faltantes en contraste con los ya descargados dentro de la dirección de almacenamiento

Si la lista de archivos faltantes tiene datos:

Bucle para iterar la lista:

Extraer el año, día y nombre de archivo de la cadena actual

Construir ruta de almacenamiento

Si la ruta no existe:

Crear la dirección completa

Llamada a la función *proDescarga*(contador, longitud de la lista iterada, enlace de descarga de archivo, nombre de archivo)

Suma +1 al contador

Sino:

Llamada a la función *proDescarga*(contador, longitud de la lista iterada, enlace de descarga de archivo, nombre de archivo)

Suma +1 al contador

Sino:

Imprimir “No hay datos para descargar”

Si la lista de direcciones de carpetas tiene datos:

Llamada al script *lectorNC* función *comprobacionNC*

Función *proDescarga*

Comienza descargar de archivo (dirección de descarga, nombre de archivo)

Script *lectorNC* función *comprobacionNC*

Bucle para iterar la lista de rutas:

Obtener lista de archivos descargados

Bucle para iterar lista de archivos descargados:

Genera una lista con archivos descargar

Bucle para iterar la lista de descargados:

Genera ruta de almacenamiento completa con adición del nombre de archivo correspondiente

Completa el enlace de descarga del archivo

Leer el archivo (NC) descargado

Si ocurre un error de archivo:

- Imprime “Archivo en mal estado”
- Archivo removido del disco
- Si ocurre un error de directorio:
 - Imprime “Ruta no existente”

Redacción de Procesos de L3

Primera parte

Entrada de datos, 2 parámetros [script_name] [path_file (cfg)]

Si el archivo no existe:

Fin del proceso

Sino: comienza proceso:

Lectura de archivo (cfg)

Creación de diccionario a partir de los archivos de configuración

Bucle para comprobar validez de los parámetros:

Si el parámetro está vacío:

Dato incorrecto

Fin

Comprobar la fecha

Si la fecha de inicio y la fecha final están vacíos:

Tomar la fecha actual para interpretarla como fecha de parada

setear en el diccionario la fecha de comienzo por defecto

Sino:

Toma las fechas del archivo de configuración

Llamada a la función “*main*” [diccionario, ruta de archivo]

Función “*main*”

Llamada a la función “*validateDate*”

Si la función retorna *verdadero*:

setear en el diccionario las fechas correspondientes en formato juliano mediante la llamada al script *formato_juliano.py* con la función “*main*”

Llamada al script *generador* con la función “*generador_enlace*” parámetros [diccionario]

Llamada al script *descargador* con la función “*descargarArchivos*” parámetros [list_url, ruta de archivo]

Script formato_juliano

Llamada a la función “*main*” parámetros

Recibe fechas:

Llamada a la función “*producto*”

Si el año es bisiesto:

Le suma un día al mes de febrero

Si el mes es enero:

Retorna una cadena con el formato juliano que le corresponde

Si el mes más de enero y menos o igual a diciembre:

Bucle para iterar una lista de los meses:

Si el mes actual es menor que diciembre:

Se suman los días totales a un contador

Retorna una cadena con el formato juliano correspondiente

Script generador

Llamada al constructor de la clase “*generado_enlace*” parámetros [diccionario]

Si el año inicial es igual al año final:

Genera url único

Sino:

Genera una lista de enlaces

Script descargador

Llamada a la función “*decargarArchivos*” parámetros [lista_url, ruta de archivo]

Bucle para iterar la lista de urls:

Genere una lista dividiendo el enlace de los años

Genera una lista de cada url obtiene el enlace a utilizar

Genere una lista con los años designados a descargar

Bucle para iterar la lista de los enlaces válidos:

Si el año actual es igual al año de inicio y fin:

Mientras un contador sea menor o igual al día final:

Condiciones a partir de la longitud del contador para generar una lista con los patrones de los días

Generar un diccionario con la url padre de los archivos y los patrones de los días de cada archivo

Sino, si el año actual es igual al año inicial y año actual es diferente al año final:

Realiza la misma instrucción

Sino, si el año actual es mayor al año de inicio y menor al año de fin:

Agrega la url padre al diccionario

Sino, si el año actual es mayor al año de inicio e igual al año de fin:

Realiza la misma instrucción que la primera condición

Llamada al script *descarga* con la función “*descarga*” parámetros [diccionario]

Script descarga

Llamada a la función “*descarga*” parámetros [diccionario]

Bucle para iterar el primer nivel del diccionario:

Bucle para iterar cada subnivel del diccionario:

Si el dato es una cadena:

Genera una lista con las urls

Sino:

Bucle para iterar los datos de lista:

Generar una lista con los datos de cada lista iterada

Bucle para iterar la lista de urls:

Si la subcadena es Annual:

Remueve la subcadena perteneciente al año

Accede a la url iterada

Si el código de respuesta es 200:

Obtiene el objeto de requests y lo pasa a código HTML texto

Bucle para recorrer el DOM en busca de etiquetas “td”:

Bucle para recorrer las etiquetas “td” en busca de anclas (a [href]):

Bucle para iterar la lista de patrones de posibles archivos:

Si la búsqueda del patrón dentro del ancla no es -1:

Genera una lista con los enlaces de los archivos

Genera un diccionario con llave de url padre igual a

los enlaces de archivos que le pertenecen

Sino:

Recibe respuesta de error

Si el código de respuesta es 200:

Bucle para iterar los elementos del diccionario:

Separa la url por cada barra “/”

Leer el archivo de configuración

Obtener ruta de almacenamiento padre

Mientras el contador sea menor a la longitud de la url separada:

Construye la ruta completa de almacenamiento (árbol de carpetas)

Si la ruta no existe:

Crea la ruta

Si la longitud de lista de datos a descargar es mayor a 0:

Bucle para recorrer la lista de descarga:

Comienza descarga del archivo actual

Si ocurren errores continúa el ciclo

Sino:

Fin

Sino:

Inspecciona la ruta en busca de archivos descargados

Bucle para iterar la lista de archivos descargados:

Genera una lista con los archivos descargados

Bucle para iterar los enlaces encontrados:

Generar lista para datos a descargar

Bucle para iterar la lista de archivos por descargar:

Si el nombre del archivo no está en los archivos descargados:

Genera una lista con los archivos faltantes

Si la longitud de los faltantes es mayor a 0:

Bucle para iterar la lista de archivos faltantes:

Comienza descarga del archivo actual

Si ocurren errores continúa el ciclo

Genera una lista con la ruta de almacenamiento de los archivos

Sino:

Fin

Si la lista de las rutas de almacenamiento es mayor a 0:

Llamada al script *lectorNC* y a la función “*comprobacionNC*” parámetros [lista con rutas de almacenamiento]

Script lectorNC

Llamada a la función “*comprobacionNC*” parámetros [lista con rutas de almacenamiento]

Bucle para iterar la lista de rutas:

Obtener lista de archivos descargados

Bucle para iterar lista de archivos descargados:

 Genera una lista con archivos descargar

Bucle para iterar la lista de descargados:

 Genere ruta de almacenamiento completa con adición del nombre de
archivo correspondiente

 Completa el enlace de descarga del archivo

 Leer el archivo (NC) descargado

Si ocurre un error de archivo:

 Archivo en mal estado

 Archivo removido del disco

 Fin

Si ocurre un error de directorio:

 Ruta no existente

 Fin