Universidad Tecnológica de la Costa

División de Ingeniería Ciencia y Tecnología

Implementación de Cadena de Descarga Automática de Datos Satelitales (Parte 1)

MEMORIA

Que para obtener el título de:

Ingeniería en Tecnologías de la Información y Comunicación

Presenta:

Brayan Alejandro De La Torre Romero

Asesor:

MCE Héctor Hugo Domínguez Jaime

Santiago Ixcuintla, Nayarit, México Abril, 2017

Dedicatorias

Mi proyecto se lo dedico especialmente a mis padres, que son la pieza esencial para estar aquí, ello con su duro esfuerzo y trabajo hicieron todo lo posible para que llegara hasta donde estoy. Siempre me brindaron su apoyo en todo lo que necesité, y agradezco la confianza que depositaron en mí.

A Sara por apoyarme y entregarme ánimos para seguir adelante, esa tierna persona que con su apoyo moral logro un gran impacto sobre mi persona, por lo cual deseo ser un gran ejemplo para ella. Para que nunca se rinda y de lo mejor que tiene en todo.

A mis compañeros de estadía que realmente fueron una pieza importante para poder lograr este objetivo, que ellos fueron quienes me ofrecieron consejos y ánimos para concluir felizmente.

Agradecimientos

En agradecimiento a la Universidad Tecnológica de la costa el haberme aceptado como uno más de sus alumnos, pero en especial a la Carrera de Tecnologías de la Información y Comunicación que fue como una familia para mí, que en momentos llegaba a pasar más tiempo debido a las largas horas de estudio.

De igual manera agradezco a los maestros que más que eso, fueron como camaradas en los ratos de ocio, en especial a mi asesor, el MCE Héctor Hugo Domínguez Jaime por aconsejarme y guiarme en los momentos que en verdad lo necesitaba.

Agradezco a mi asesor empresarial, el Dr. Juan Pablo Rivera Caicedo por haberme otorgado la oportunidad de realizar mi estadía bajo su cargo y por darme una nueva perspectiva de los diversos enfoques que tiene la carrera de sistemas informáticos.

Índice

Índice de tablas y/o figuras

Resumen

El laboratorio de percepción remota en CENIT actualmente se encarga de realizar estudios sobre cuerpos de agua, vegetales y forestales, oceanográficos y zonas costeras, entre otras más. Dicho esto, la manera de obtener los datos satelitales se realiza como un proceso manual, lo que se traduce a largos periodos de tiempo en caso de necesitar gran cantidad de datos, por este motivo se requiere agilizar el proceso de descarga de los datos de manera dinámica mediante la implementación de una cadena de procesado ejecutada como script, es decir, no es necesario implementar plataformas gráficas, todo debe ser ejecutado en un proceso como tal. El script debe ser capaz de obtener datos y generar consultas a partir de los parámetros proporcionados. Con ello la descarga de datos será tan sencillo como especificar los datos que son requeridos en el momento. Todo este desarrollo se hizo con el modelo Scrum, el cual nos permite realizar el proceso por hitos y demostrar avances manteniendo cada proyección documentada dentro del mismo modelo. Con ello concluyo que la cadena de proceso que se encargará de descargar los datos de manera automática, realizará la tarea de descargar los datos, pero antes de ello deberá verificar la existencia para evitar la descarga de datos anteriormente descargados, además de almacenarlos en el árbol de directorios que le corresponde a cada archivo que se esté obteniendo en ese momento, y por último, verificar si los datos han sido descargados correctamente, evitando así fallos de proceso posterior a la descarga de los mismo.

Capítulo 1. Planteamiento del problema

1.1 Antecedentes

El CENIT es un centro de alto nivel que pone a disposición la capacidad científica y tecnológica instalada para atender a través de investigaciones de gran calidad los problemas de acuacultura, la ganadería, los problemas del campo, del desarrollo económico, del turismo y del medio ambiente. Por parte del Laboratorio de Percepción Remota, se realizan investigaciones oceanográficas, donde por medio de datos satelitales, se llevan a cabo procesos de medición cuantitativa de diversos estudios.

La descarga de los datos se realiza de forma manual, donde el usuario normalmente ingresa a la plataforma dentro de un sitio web; proporcionado por OCEANCOLOR en relación con la NASA, los usuarios ingresan a la plataforma según sea la información que desean obtener. Una vez localizados los datos de interés, procede la acción de descargar los datos de forma individual, seleccionar uno y descarga, seguido por otro y de igual forma.

Surge la necesidad de llenar un base de datos con hasta un año de información, los usuarios deben descargar estos archivos uno a uno hasta completar el ciclo anual de datos.

La descarga manual de esta información se demora bastante y, además de ser tedioso el trabajo repetitivo que se debe realizar, surge la mayor necesidad de automatizar este proceso.

Cada parámetro de una consulta de datos se construye de la siguiente manera: SENSOR (satélite/plataforma), TIPO DE PRODUCTO (L2, L3, mapeado), PERIODO DE DATOS (desde un paquete diario hasta paquetes de información anual), RESOLUCION (km/px), TIPO DE ESTUDIO (clorofila, temperatura, etc.) y FECHA (hasta el día de datos requeridos). Donde cada parámetro de esta consulta se deriva en un directorio, por lo que al final para obtener los datos se debe recorrer un árbol de directorios dentro de una plataforma web de cada sensor.

Cada consulta a la plataforma web está relativamente ordenada en capas de directorios, hacer un sinfín de clics para obtener un solo archivo resulta en un proceso bastante molesto.

Después de un largo periodo de descarga, el proceso es algo que se realiza de forma de igual forma, manualmente. Sin embargo el problema primordialmente reside en el proceso de la descarga.

Aunque esto suene redundante, esta labor de realizarse de la misma forma repercute a los procesos seguidos, mientras más demorado sea obtener los datos, el procesado de los mismos se ralentiza aún más.

1.2 Objetivos

Desarrollar un script para la descarga automática de datos satelitales alojados en una página web. Responsable de la NASA Oficial: Gene C. Feldman (https://oceancolor.gsfc.nasa.gov/).

1.2.1 Objetivos específicos

- Desarrollar un script para la descarga automática de datos de Nivel 3, satelitales alojados dentro la plataforma OCEANCOLOR utilizando el lenguaje Python
- Desarrollar un script para la descarga automática de datos de Nivel 2, satelitales alojados dentro la plataforma OCEANCOLOR utilizando el lenguaje Python

1.3 Justificación

La principal razón por la cual el Laboratorio de Percepción Remota (PERSEO), realiza investigaciones sobre las zonas, terrestres, oceanográficas y medioambientales es para determinar el estado en el que se encuentra el planeta Tierra.

El proceso de descarga automática permitirá a los usuarios realizar la tarea en cuestión de poco tiempo, reduciendo el tiempo de recolección de datos significativamente.

El script que se requiere contará con una serie de parámetros, en los cuales se apoyara para construir las consultas que deberá procesar la plataforma web donde se alojan los datos. Por medio de archivos externos se pueden configurar estos parámetros, haciéndolo de esta forma, versátil y de cómoda usabilidad.

Al ser un archivo de ejecución individual, permite realizar más de una tarea al mismo tiempo lo que agiliza aún más la descarga de múltiples datos.

Para que el script pudiera ser capaz de realizar tal orden de manera perfecta, se debió estudiar de manera detallada; localmente los posibles parámetros que construirían una consulta, ya que si alguno de estos está errado, el proceso terminaría antes de comenzar, por otro lado externamente, la plataforma web que pone a disposición los datos de manera pública, esto se debe a que cada consulta, dependiendo del tipo de producto que se necesita, cambiara ya sea una mínima parte o totalmente en su compasión y destino de procesamiento, es decir, a que plataforma se ha de dirigir la consulta.

De esta manera, la cadena de descarga automática de datos satelitales nace, para satisfacer las necesidades de obtener gran cantidad de datos, de así ser requerido, en cuestión de minutos, para asistir en la agilización de las investigaciones realizadas dentro de PERSEO.

1.4 Limitaciones

Lo primero a desarrollar será el conjunto de scripts para automatizar la descarga de los datos de nivel 3, partiendo primeramente por generar la consulta dinámica que se enviará a la página donde se alojan todos esos datos. Seguidamente, se especificarán las instrucciones que debe realizar el script para obtener los datos que son requeridos por cada ejecución del script principal. Por último se debe integrar la cadena de procesado dentro de un sólo script, el cual será ejecutado cada que se le requiera.

Por consiguiente, el desarrollo del script para la descarga automática de los datos de nivel 2, de igual manera, el punto de partida es generar una consulta dirigida a la web, que contendrá los parámetros necesarios para obtener datos específicos, después de ello, es necesario generar nuevas consultas por cada parte de información obtenida en la primera consulta, lo que sigue es una nueva consulta, ésta se encargará de obtener los patrones de los datos que son necesitados. Para finalizar, la cadena de procesos debe ser integrada en un script principal, el cual será ejecutado para los datos de Nivel 2.

Capítulo 2. Marco Teórico

2.1 Fundamentos

2.1.1 El producto

Para Jesús González Bernal (2017) “La información adquirida con detectores a bordo de satélites o aviones ha resultado de gran utilidad para muchas y diferentes aplicaciones: agricultura, minería, fenómenos naturales, detección de aguas contaminadas; asimismo para el monitoreo de bosques y de glaciares, entre otras”.

“La clorofila prefiere la luz azul a la verde, o sea que absorbe más fácilmente la parte más energética del espectro visible del sol (el azul). Es por eso que la clorofila y todas las plantas en general son verdes, al ser éste el color que reflejan” (Anne Cabrés Albós, 2014).

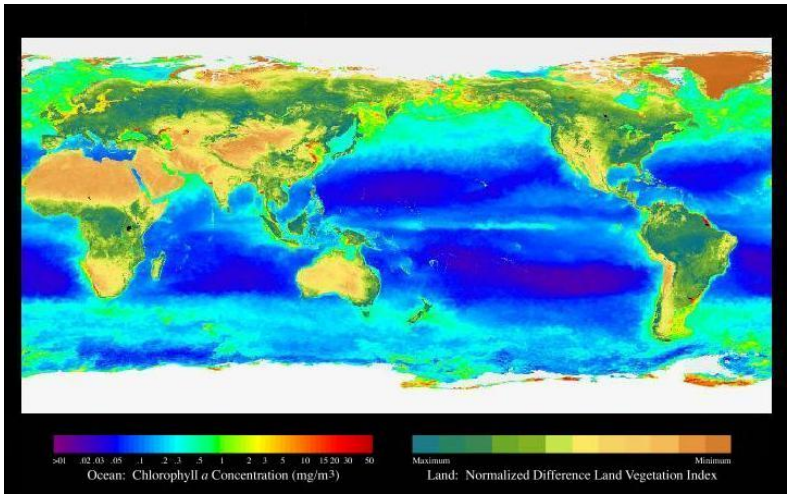


Figura 1. Distribución anual media de clorofila en todos los océanos

“Cada dos horas, el satélite da una vuelta alrededor de la Tierra de norte a sur pasando por los dos polos (aproximadamente). Durante estas dos horas rastrea dos franjas horarias, siempre pasando por el ecuador a las 12 del mediodía. Al cabo de dos horas, la Tierra ha girado

dos horas y, sin moverse, el satélite puede empezar a rastrear las dos franjas horarias siguientes otra vez pasando por el ecuador a las 12 del mediodía de la siguiente franja. En resumen, a este satélite le cuesta 24 horas (un día) rastrear toda la Tierra" (Anne Cabrés Albós, 2014).

La limpieza de datos. Sólo un 10% de la luz detectada es proveniente de la superficie de los océanos, mientras que el resto pertenece a la atmosfera, además de las nubes. Por lo tanto se debe filtrar todo el 'ruido' de la atmosfera y otros elementos naturales que se encuentren allí mismo. El resultado, un mapa con el color del océano. Después de ello a descifrar la cantidad de clorofila, basado en el principio de que la clorofila absorbe mucha radiación solar hacia el azul (Anne Cabrés Albós, 2014).

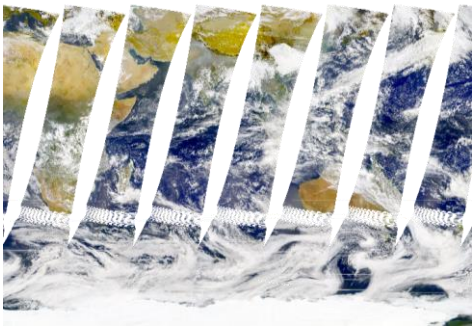


Figura 2. Trozo de mapa de lo que el satélite vio antes de tratar los datos

"En primavera, la temperatura empieza a subir, y las capas más superficiales del océano se estratifican (menos turbulencia), así que el fitoplancton puede vivir en las capas bien iluminadas sin ser continuamente arrastrado para el fondo. Es el momento ideal para su supervivencia. Después de consumir todos los nutrientes acumulados durante el invierno, esta explosión de fitoplancton muere a los pocos meses" (Anne Cabrés Albós, 2014).

“el fitoplancton absorbe CO2 a través de la fotosíntesis y lo convierte en carbono orgánico. Después, todos los depredadores se comen a otros depredadores que se comen el fitoplancton, y las heces trasladan un pequeño porcentaje de este carbono al fondo del mar, donde sedimenta, y queda lejos de la atmósfera por miles de años” (Anne Cabrés Albós).

2.1.2 Sensores

“Las actividades humanas han cambiado drástica y rápidamente la cobertura de nuestro planeta” (Tzitziki Janik García-Mora; Jean-François Mas, 2011).

“Las actividades humanas han cambiado drásticamente la cobertura de nuestro planeta: los cultivos y pastizales se han expandido ocupando cerca del 40% de la cobertura mundial del suelo. Hace ya varias décadas se puso de manifiesto que los cambios de cobertura y de uso de suelo influyen directamente en los ciclos hidrológicos, la pérdida de biodiversidad, la erosión de los suelos y el aumento de gases que incrementan el efecto invernadero. Los cambios climatológicos provocan el aumento y la intensidad de los desastres naturales en todo el planeta como son los incendios, las inundaciones, los huracanes y las sequías” (FAO, 2007).

Los sensores MODIS (Aqua y Terra) dentro de los satélites TERRA y AQUA realizan un papel importante en el estudio de la Tierra; los comportamientos superficiales oceánicos y atmosféricos, todo para desarrollar modelos que sean capaces de predecir el cambio global que ocurre en el planeta Tierra, y se puedan tomar acciones referentes a la protección medioambiental (CREPAD, s/f).

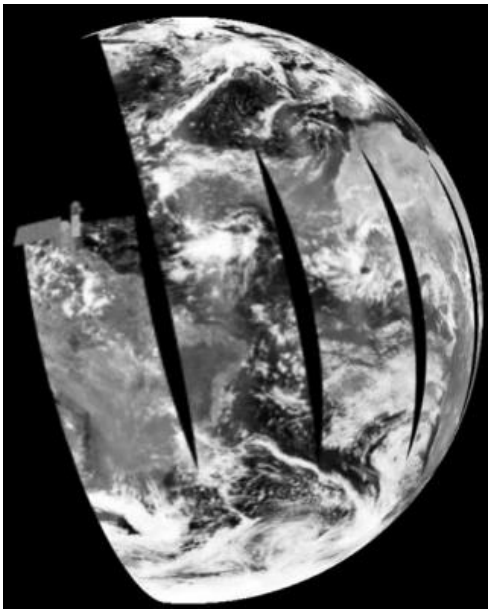


Figura 3. Sistema de escaneo del sensor MODIS

"Junto con todos los datos de otros instrumentos a bordo de Terra y Aqua, los datos MODIS se transfieren a las estaciones en tierra en White Sands, Nuevo México, a través del seguimiento y retransmisión de datos desde el sistema de satélites (Tracking and Data Relay Satellite System -TDRSS)" (Tzitziki Janik García-Mora; Jean-François Mas, 2011).

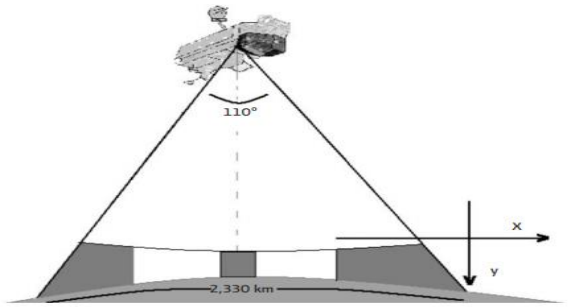


Figura 4. Ángulo de visión del sensor MODIS

2.1.3 Orbitas

"Cuando se quiere realizar un estudio sobre una zona geográfica pequeña para caracterizar, por ejemplo, la distribución de determinados tipos de vegetación o suelos en una zona, se suele comparar la información espectral medida desde el satélite con información medida *in situ* con un espectro-radiómetro de campo" (Daniel Rodríguez Pérez, 2015).

2.1.4 Producto por niveles

Los productos que ofrecen los sensores MODIS se dividen en cinco diferentes niveles, según esto, sea el grado de procesamiento (Tzitziki Janik García-Mora; Jean-François Mas, 2011).

"La utilización del sensor MODIS ha dejado una gran experiencia y conocimiento en la comunidad científica, y debido a que el final de sus operaciones se acerca ya se prepara el sensor

que dará continuidad a este tipo de datos" (Tzitziki Janik García-Mora; Jean-François Mas, 2011).

"Las plataformas Terra y Aqua cuentan con varios sensores, lo que representa grandes ventajas al obtener diferentes tipos de información con las mismas condiciones atmosféricas, de ángulo de iluminación y de observación y permitiendo la inter-calibración entre sensores" (Tzitziki Janik García-Mora; Jean-François Mas, 2011).

2.2 Python

2.2.1 Algo sobre Python

"Python es un lenguaje de programación de alto nivel cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación" (David, 2015).

Elementos del lenguaje de programación Python

"Python fue diseñado para ser leído con facilidad. Entre otras cosas se utilizan palabras en inglés donde otros lenguajes utilizarían símbolos" (David, 2015)

2.3 Peticiones

"HTTP contiene un grupo de peticiones HTTP (también llamadas HTTP verbs por, el tipo de nombre que manejan casi todos ellos -pues si bien algunos son sustantivos, la gran mayoría no-) que nos ayudan a especificar la acción que se requiere realizar en un elemento determinado

y aunque estas peticiones tienen distintas semánticas, también tienen muchas similitudes en las mismas que evitan que este grupo se extienda demasiado" (Mangel, 2018).

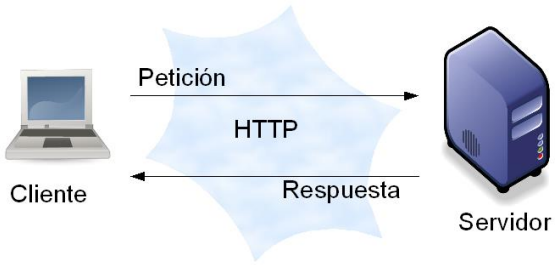


Figura 5. Esquema de peticiones HTTP

2.4 Almacenamiento de datos

Base de datos vs directorio

"Además del espacio y la rapidez de acceso, creo que deberías evaluar qué tan importante es la imagen para la entidad de negocio. Si la imagen es crítica en cuanto a su pérdida, deberías persistirla dentro de la tabla como array de byte. Imaginemos una entidad que representa un producto y tienes las imágenes de sus planos de construcción, en este caso si algún error hiciera que la carpeta de imágenes se perdiera sería crítico, en este caso la imagen debe formar parte de la entidad en la tabla. Ahora si tengo una entidad persona en donde define su perfil una foto, si algo hiciera que se perdiera no sería crítico, se podría solicitar que actualice su foto en caso de hacer falta, aquí se podría ubicar la foto en una carpeta" (Leandro Tuttini, 2017).

"El BLOB cuanto menos se use mejor. No solo hace la tabla bastante más grande (y consecuentemente más lenta), si no que para acceder al fichero requiere hacer el parseo. Yo recomiendo solo poner la referencia. Si los ficheros son públicos (imágenes de un blog o algo así) colocándolo en una carpeta pública y configurando bien Apache / Nginx el acceso es directo, siendo más liviano y rápido. Si por el contrario es de acceso restringido, lo colocas en una

carpeta privada y que el código servidor, tras comprobar si hay permiso, acceda al fichero y los "escupa" tal cual" (Miguel Ángel López Vicente, 2014).

2.5 Web scraping

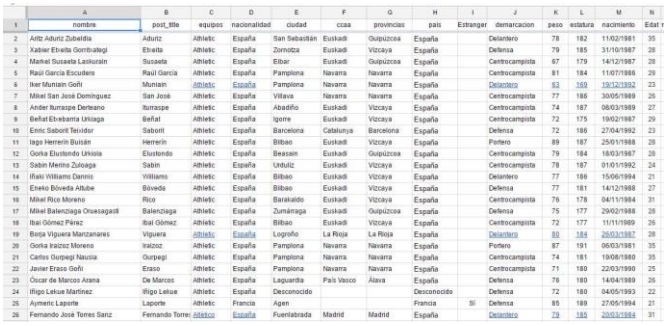
"Podemos aprovechar el web scraping para conseguir cantidades industriales de información (Big data) sin teclear una sola palabra. A través de los algoritmos de búsqueda podemos rastrear centenares de webs para extraer sólo aquella información que necesitamos" (Marq Martí, 2016).

"El web scraping es una disciplina que debe combinar dos vertientes muy diferenciadas del conocimiento web, ambas esenciales para tener perfil versátil en la red. Por un lugar debemos dominar la visualización de datos a nivel conceptual y por el otro debemos disponer de los conocimientos técnicos necesarios para lograr extraer con exactitud los datos con herramientas especializadas" (Marq Martí, 2016).

"Al fin y al cabo esto se resumirá en saber gestionar grandes cantidades de datos (big data). Debemos estar mínimamente familiarizados con la visualización de grandes cantidades de datos con tal de poder jerarquizar e interpretar los datos que extraigamos de una web. Y no solo para extraer los datos, también a la hora de plantear la estrategia de extracción debemos saber cuáles van a ser los datos que vayamos a extraer con tal de poder darles un sentido informativo para el usuario" (Marq Martí, 2016).

Dentro del concepto se definen tres puntos que se deben manejar al momento de realizar web scraping: los conocimientos de maquetación, utilización de visualizadores de datos y conocimientos solidos de expresiones regulares (regex). (Marq Martí, 2016)

"El web scraping consiste en obtener los datos pero evidentemente estos datos los tendremos que usar para alguna finalidad. Es aquí cuando entran en juego dos procesos claves una vez obtenidos los datos: Jerarquización, ordenación y filtrado de los datos, e Importación de los datos a otra plataforma" (Marq Martí, 2016)



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	nombre	post_title	equipos	nacionalidad	ciudad	ccaa	provincias	pais	Estranger	demarcacion	peso	estatura	nacimiento	Edat r
2	Aritz Aduriz Zubeldia	Aduriz	Athletic	España	San Sebastián	Euskadi	Guipúzcoa	España		Delantero	78	182	11/02/1981	35
3	Xabier Etxeita Gorritxategi	Etxeita	Athletic	España	Zornotza	Euskadi	Vizcaya	España		Defensa	79	185	31/10/1987	28
4	Markel Susaeta Laskurain	Susaeta	Athletic	España	Eibar	Euskadi	Guipúzcoa	España		Centrocampista	67	179	14/12/1987	28
5	Raúl García Escudero	Raúl García	Athletic	España	Pamplona	Navarra	Navarra	España		Centrocampista	81	184	11/07/1986	29
6	Iker Muniain Goñi	Muniain	Athletic	España	Pamplona	Navarra	Navarra	España		Delantero	63	169	19/12/1992	23
7	Mikel San José Domínguez	San José	Athletic	España	Villava	Navarra	Navarra	España		Centrocampista	77	186	30/05/1989	26
8	Ander Iturraspe Derteano	Iturraspe	Athletic	España	Abadiño	Euskadi	Vizcaya	España		Centrocampista	74	187	08/03/1989	27
9	Beñat Etxebarria Urkiaga	Beñat	Athletic	España	Igorre	Euskadi	Vizcaya	España		Centrocampista	72	175	19/02/1987	29
10	Enric Saborit Teixidor	Saborit	Athletic	España	Barcelona	Catalunya	Barcelona	España		Defensa	72	186	27/04/1992	23
11	Iago Herrerín Buisán	Herrerín	Athletic	España	Bilbao	Euskadi	Vizcaya	España		Portero	89	187	25/01/1988	28
12	Gorka Elustondo Urkiola	Elustondo	Athletic	España	Beasain	Euskadi	Guipúzcoa	España		Centrocampista	79	184	18/03/1987	28
13	Sabin Merino Zuloaga	Sabin	Athletic	España	Urduliz	Euskadi	Vizcaya	España		Centrocampista	78	187	01/01/1992	24
14	Íñaki Williams Damnis	Williams	Athletic	España	Bilbao	Euskadi	Vizcaya	España		Delantero	77	186	15/06/1994	21
15	Eneko Bóveda Altube	Bóveda	Athletic	España	Bilbao	Euskadi	Vizcaya	España		Defensa	77	181	14/12/1988	27
16	Mikel Rico Moreno	Rico	Athletic	España	Barakaldo	Euskadi	Vizcaya	España		Centrocampista	76	178	04/11/1984	31
17	Mikel Balenziaga Oruesagasti	Balenziaga	Athletic	España	Zumárraga	Euskadi	Guipúzcoa	España		Defensa	75	177	29/02/1988	28
18	Ibai Gómez Pérez	Ibai Gómez	Athletic	España	Bilbao	Euskadi	Vizcaya	España		Centrocampista	72	177	11/11/1989	26
19	Borja Viguera Manzanares	Viguera	Athletic	España	Logroño	La Rioja	La Rioja	España		Delantero	80	184	26/03/1987	28
20	Gorka Iraizoz Moreno	Iraizoz	Athletic	España	Pamplona	Navarra	Navarra	España		Portero	87	191	06/03/1981	35
21	Carlos Gurpegi Naustia	Gurpegi	Athletic	España	Pamplona	Navarra	Navarra	España		Centrocampista	74	181	19/08/1980	35
22	Javier Eraso Goñi	Eraso	Athletic	España	Pamplona	Navarra	Navarra	España		Centrocampista	71	180	22/03/1990	25
23	Óscar de Marcos Arana	De Marcos	Athletic	España	Laguardia	País Vasco	Álava	España		Defensa	78	180	14/04/1989	26
24	Iñigo Lekue Martínez	Iñigo Lekue	Athletic	España	Desconocido			Desconocido		Defensa	72	180	04/05/1993	22
25	Aymeric Laporte	Laporte	Athletic	Francia	Agen			Francia	Sí	Defensa	85	189	27/05/1994	21
26	Fernando José Torres Sanz	Fernando Torres	Atlético	España	Fuenlabrada	Madrid	Madrid	España		Delantero	79	185	20/03/1984	31

Figura 6. Hoja de Google Spreadsheets con datos extraídos con un web scrapper de las webs de laliga.es y uefa.com.

2.6 REGEX (expresiones regulares)

"Las expresiones regulares están disponibles en casi cualquier lenguaje de programación, pero aunque su sintaxis es relativamente uniforme, cada lenguaje usa su propio dialecto" (Irv, 2005).

"Se trata sencillamente de ir cotejando un patrón (pattern) con una cadena (subject) y ver si dentro de ella existe la misma secuencia. Si existe, decimos que hemos encontrado una coincidencia (match, en inglés)" (Irv, 2005).

```
<? am // este es nuestro patrón. Si lo comparamos con:
am // coincide
panorama // coincide
ambicion // coincide
campamento // coincide
mano // no coincide
?>
```

Figura 7. Explicación básica de cómo se comporta una expresión regular

2.7 Archivos de registro de procesos informáticos

"Cuando trabajas frente al ordenador, navegas en tu tablet u operas una página web desde un servidor, tienen lugar numerosos procesos que pasan inadvertidos ante cualquier usuario. En caso de que se presenten problemas, se produzcan errores o quieras conocer exactamente qué acciones ejecutan los sistemas operativos o los diferentes programas o servicios, puedes acceder a los llamados archivos log. Estos "logs" son gestionados por prácticamente todas las aplicaciones, servidores, bases de datos y sistemas de manera automática y permiten controlar (de forma centralizada) todos los procesos relevantes" (Digital Guide, 2016).

"Como consecuencia del registro detallado de datos de los logs, estos son una fuente primordial a la hora de analizar errores de programa o del sistema, así como para determinar el comportamiento de los usuarios" (Digital Guide, 2016).

"Los sistemas operativos crean múltiples logs en los que se registran y clasifican diferentes tipos de procesos. Los sistemas Windows realizan protocolos sobre eventos de aplicaciones (programas), eventos del sistema, eventos relacionados con la seguridad, eventos de configuración y eventos reenviados. Consultar la información contenida en un log puede ayudar a un administrador a solucionar problemas" (Digital Guide, 2016).

"Aunque aún se practica el análisis de logs de un servidor web, la aparición de nuevos métodos de análisis web como Cookies o Page Tagging lo han desplazado en gran parte. Las razones para ello son, en primer lugar, la tasa alta de error del análisis de archivos log en la asignación de sesiones y, en segundo lugar, el hecho de que el propietario de una página web muy rara vez puede acceder a los archivos de registro de un servidor web" (Digital Guide, 2016)

Capítulo 3. Metodología

3.1 Planeación

3.1.1 Grafica de Gantt

Comentado [UdMO1]: Si esta es la grafica de gantt del proyecto, hay que ponerla en una hoja sola, para apreciar las etapas o las fases del proyecto.

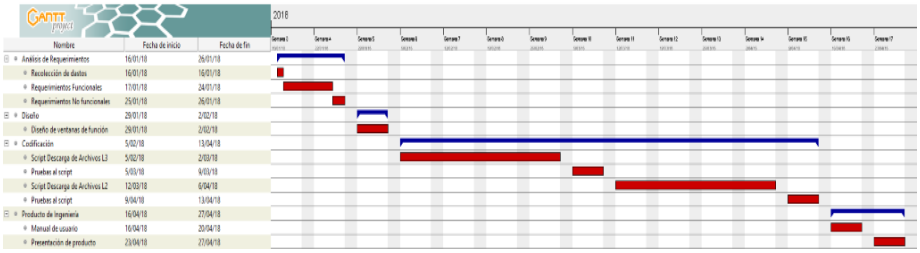


Figura 8. Gráfica de Gantt

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 8 muestra las actividades y tiempo de duración de cada una; con el cual se trabajara en este proyecto.

Así mismo la figura 9 a manera de una lista nos muestra las actividades que conlleva el desarrollo del proyecto mostrando los intervalos de duración para cada uno.

Comentado [UdMO2]: El pie de la imagen, cuadro o foto, debe ir centrado, al igual que la foto, imagen o tabla.

Cadena de Descarga Automatica de Datos Satelitales 12-mar-2018

Tarea 2

Nombre	Fecha de inicio	Fecha de fin
Análisis 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 9. Agenda de actividades del proyecto

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.

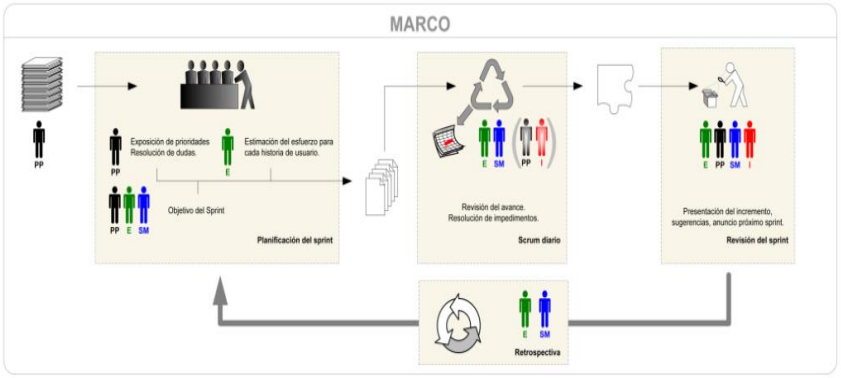


Figura 10. Esquema del modelo de desarrollo Scrum

3.2 Fase de Análisis

3.2.1 Técnica de recolección de datos

La entrevista supone más un conversación cuya finalidad es la de obtener información relevante para estimar resultados cualitativos.

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?

   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, asi 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

    - 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

3.2.2.1 Funcionales

- Descargar archivos de nivel 3 (L3) de la plataforma OCEANCOLOR.
- Descargar archivos de nivel 2 (L2) de la plataforma OCEANCOLOR.
- 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 defecto).

3.2.2.2 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 Diseño

En esta fase se describen los diagramas de cada subproceso realizado para la cadena de procesamiento principal.

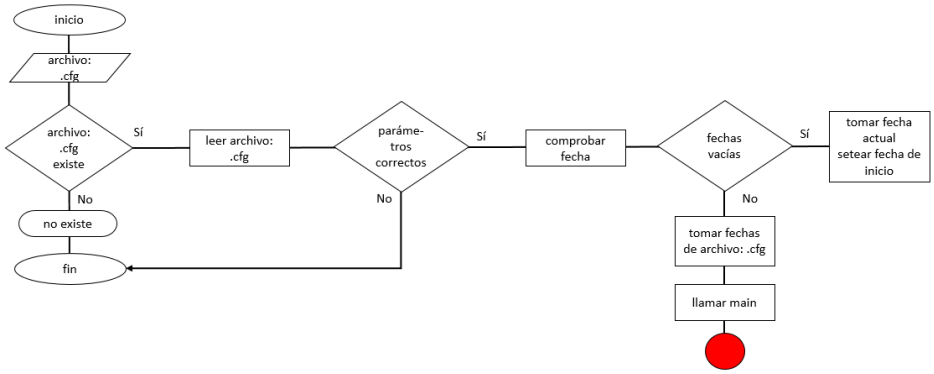


Figura 11. Diagrama de proceso principal subscript "main.py"

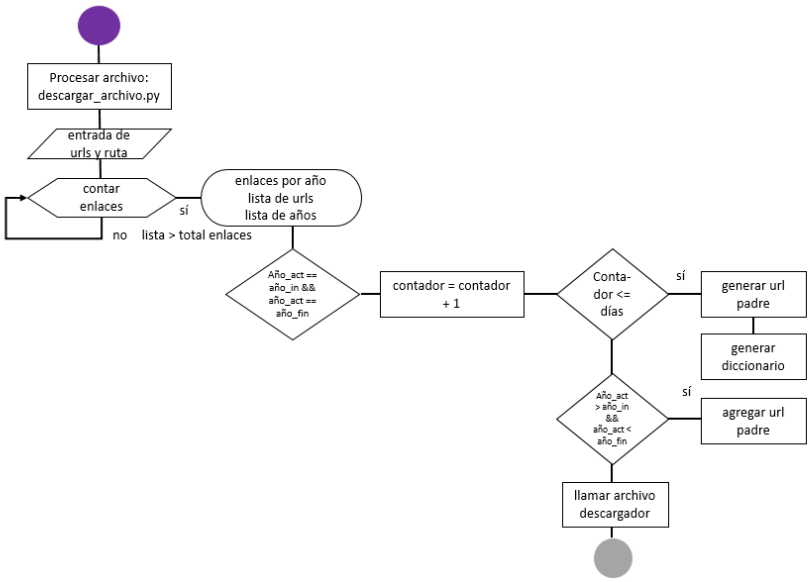


Figura 12. Diagrama de subproceso de descarga de archivo

3.4 Codificación

```
def main(fecha):
    dates = []
    try:
        fecha = fecha.split('-')

        for x in fecha:
            d = datetime.strptime(x, '%d/%m/%Y')

            d = datetime.strftime(d, '%d/%m/%Y')

            year = int(d[-4:])

            month = int(d[3:5])

            day = int(d[0:2])

            dates.append(producto(year,month,day))
        return dates
    except:
        print 'Formato de fecha no soportado'

def producto(year,month,day):
    meses = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]

    if calendar.isleap(year):
        meses[1]+=1

    nom_mes = ['enero', 'febrero', 'marzo', 'abril', 'mayo', 'junio', 'julio', 'agosto','septiembre', 'octubre', 'noviembre', 'diciembre']
    c = 0
    if month == 1:
        if len(str(day)) == 2:
            return str(year)+'0'+str(day)
        elif len(str(day)) == 1:
            return str(year)+'00'+str(day)
    elif month > 1 and month < 13:
        #print nom_mes[(month-2)]
        #print str(year)+str(meses[month-2]+day)
        for x, value in enumerate(meses):
            if x < month-1:
                c += value
                #print c

        if len(str(c)) == 2:
            return str(year)+'0'+str(c+day)
        else:
            return str(year)+str(c+day)
```

Figura 13. Script formato_juliano

Este script se encargar de generar un formato, obteniendo como argumento un par de fechas, ambas separadas por el signo guion (-). El formato de salida para cada fecha se le conoce como formato de fecha Juliano, donde el mes es interpretado en total de días que éste tiene. Ejemplo: 01/01/2000 ≈ 2000001; 01/02/2000 ≈ 2000032 ← si bien enero tiene 31, siendo 1ro del mes siguiente, lo que sucede en el formato juliano es que, al día se suman los días totales del mes anterior, siempre y cuando estén dentro del mismo año. Por lo tanto tenemos que 31 (días de enero) + 1 (primer día de febrero) = 32.

La figura 12 muestra una expresión detallada de los días julianos dentro de un año.

mes/día	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Enero	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Febrero	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59			
Marzo	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
Abril	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	
Mayo	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151
Junio	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	
Julio	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212
Agosto	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243
Septiembre	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	
Octubre	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304
Noviembre	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	
Diciembre	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365

año bisiesto

mes/día	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Enero	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Febrero	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60		
Marzo	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91
Abril	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	
Mayo	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152
Junio	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	
Julio	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213
Agosto	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244
Septiembre	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	
Octubre	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305
Noviembre	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	
Diciembre	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366

Figura 14. Conteo de los días julianos.

```
def validateDate(fecha):
    from datetime import datetime
    try:
        fecha = fecha.split('-')
        if len(fecha) == 2:
            hoy = datetime.strftime(datetime.now().date(), '%d/%m/%Y')
            hoy = datetime.strptime(hoy, '%d/%m/%Y')
            fecha_inicio = datetime.strptime(fecha[0], '%d/%m/%Y')
            fecha_fin = datetime.strptime(fecha[1], '%d/%m/%Y')

            if str(fecha_inicio) > str(fecha_fin):
                print 'La fecha inical no debe sobrepasar a la fecha de fin'
                return False
            if str(fecha_inicio) > str(hoy) or str(fecha_fin) > str(hoy):
                print 'La fecha sobrepasa a la presente'
                return False
        else:
            print '\nFormato de fecha incorrecto\n'
            return False

        return True
    except TypeError:
        print '\nFormato de fecha no soportado\n'
    except ValueError:
        print '\nFormato de fecha incorrecto\n'
    except:
        raise

    return False

def validar(file):

    from ConfigParser import SafeConfigParser

    reader = SafeConfigParser()
    reader.read(file)

    sensor = reader.get('PARAMETROS', 'sensor')
    periodo = reader.get('PARAMETROS', 'periodo').lower()
    resolucion = reader.get('PARAMETROS', 'resolucion').lower()
    producto = reader.get('PARAMETROS', 'producto').lower()

    fi = reader.get('PARAMETROS', 'fecha_inicial')
    ff = reader.get('PARAMETROS', 'fecha_final')

    # listas
    sensores = ['MODIS-Aqua', 'MODIS-Terra', 'VIIRS']
    ################################################
    periodos = ['8day', '8d_climatology', 'annual', 'bimonthly_1', 'bimonthly_2', 'cumulative', 'daily', 'monthly', 'monthly_climatology', 'rolling_32_day', 'rolling_quicklook'
        , 'seasonal', 'seasonal_climatology']
```

Figura 16. Script validacion

El proceso de descarga de los archivos de nivel 3 proporcionados por NASA desde su sitio web OCEANCOLOR se maneja mediante un archivo de configuración *ver Figura 14*. Este script se encarga de revisar cada parámetro contenido dentro del tal archivo, para que el proceso funciones adecuadamente y el resultado sea el que se espera.

```
config.cfg
[RUTA]
ruta = "C:\Users\Usuario-03\Documents\"

[PARAMETROS]
sensor = MODIS-Aqua
periodo = 8day
resolucion = 4km
producto = chl_gsm
fecha_inicial = 01/01/2017
fecha_final = 20/01/2017
```

Figura 15. Parámetros de archivo de configuración para descarga de datos L3

Capítulo 4. Resultados

La descarga de los datos satelitales resulto exitosa tras diversas complicaciones en pruebas, debido al proceso, éste no debía irrumpir en la seguridad del servidor que provee toda la información necesaria para finalizar la tarea asignada.

Después de altas correcciones en código, llegamos al resultado esperado; la descarga automática de los datos funciona como debería, tal como fue descrito al inicio de estadías.

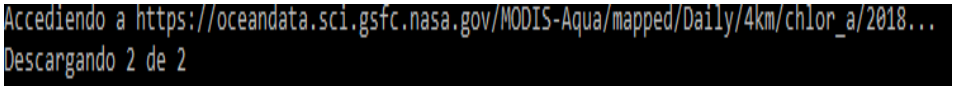


Figura 17. Ejecución de script de descarga de datos L3

En la figura 15 se muestra el resultado de la ejecución de este script, la descarga de los datos se efectuó exitosamente.

Por otro lado, la descarga es el proceso prioritario, que fue definido como tal en los objetivos del proyecto. Sin embargo esto por sí solo no funcionaría adecuadamente, teniendo en cuenta que se trabaja con usuarios, resumiendo todo, el usuario puede cometer errores que resulten fatales para el proceso.

Por lo tanto, se realizaron desarrollos para evitar desastres en la ejecución del proceso principal.

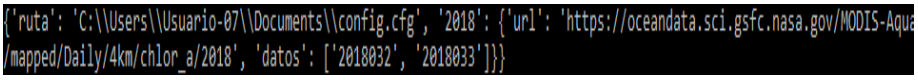


Figura 18. Ejecución de script descargador

Este script se encarga de generar la información que necesita el script de descarga.

En cuanto a los subprocesos realizados durante la ejecución de la descarga de datos de nivel 3.

Está el script para convertir las fechas en formato de días julianos, esto con la finalidad de encontrar datos dentro de ciertas fechas.

El resultado obtenido es tal como lo muestra la figura 19.

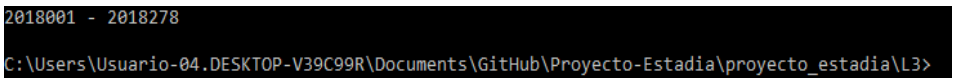
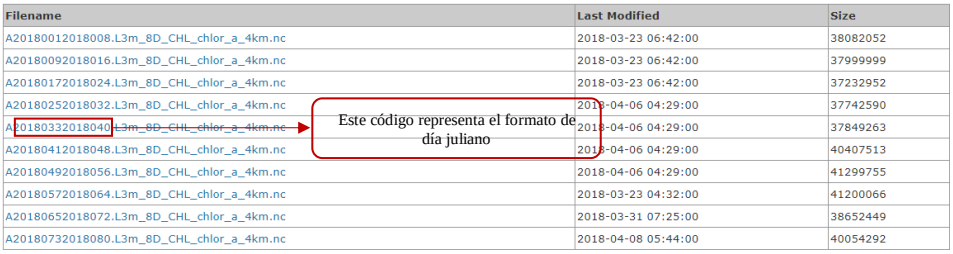


Figura 19. Resultado proceso de conversión de fechas a formato de día juliano

Este script forma parte de un formato que es requerido al momento de identificar archivos que serán descargados, adaptándose al entorno dentro de la plataforma OCEANCOLOR.



Filename	Last Modified	Size
A20180012018008.L3m_8D_CHL_chlor_a_4km.nc	2018-03-23 06:42:00	38082052
A20180092018016.L3m_8D_CHL_chlor_a_4km.nc	2018-03-23 06:42:00	37999999
A20180172018024.L3m_8D_CHL_chlor_a_4km.nc	2018-03-23 06:42:00	37232952
A20180252018032.L3m_8D_CHL_chlor_a_4km.nc	2018-04-06 04:29:00	37742590
A20180332018040.L3m_8D_CHL_chlor_a_4km.nc	2018-04-06 04:29:00	37849263
A20180412018048.L3m_8D_CHL_chlor_a_4km.nc	2018-04-06 04:29:00	40407513
A20180492018056.L3m_8D_CHL_chlor_a_4km.nc	2018-04-06 04:29:00	41299755
A20180572018064.L3m_8D_CHL_chlor_a_4km.nc	2018-03-23 04:32:00	41200066
A20180652018072.L3m_8D_CHL_chlor_a_4km.nc	2018-03-31 07:25:00	38652449
A20180732018080.L3m_8D_CHL_chlor_a_4km.nc	2018-04-08 05:44:00	40054292

Figura 20. Forma en que MODIS nombra los archivos procesados

El proceso de descarga de datos se enfoca en ubicar la web donde se alojan dichos datos, por ello se desarrolló un subproceso que genera dinámicamente estos enlaces a los sitios necesarios.

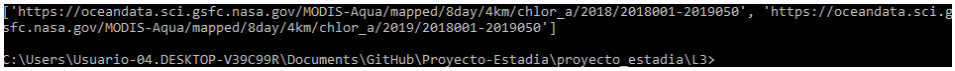


Figura 21. Subproceso de generación de enlaces dinámicos

Como es de suponer, al trabajar con usuarios, estos procesos están sujetos a errores, por ello, se opta por utilizar un control específico, es decir, un archivo de configuración donde se especifican todos los parámetros que cada proceso necesita. Sin embargo esto no es del todo

viable, sin lugar a duda pueden ocurrir algunos errores por lo que desarrollar un subproceso que encargue de validar estos parámetros.

Figura 22. Subproceso para verificar utilidad de los parámetros

Al finalizar la descarga de los datos, estos deben ser corroborados, asegurar su perfecto estado, por ello, un nuevo subproceso fue desarrollado, que se encarga de comprobar el estado de cada archivo previamente descargado.

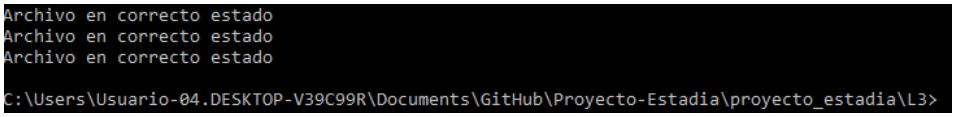


Figura 23. Subproceso para verificar el estado de los archivos descargados

Todos estos subprocesos vienen comandados por un script principal, este proceso se reconoce como como la raíz de la cadena de descarga. Este se encarga de realizar ciertas tareas de definición y llamadas a los demás subprocesos cuando se les sea requerido.

Este script recibe dos parámetros para su inicio: primeramente, mencionar que para ejecutar el script, el usuario debe situarse en la carpeta donde reside tal archivo mediante el uso de la consola MS-DOS.

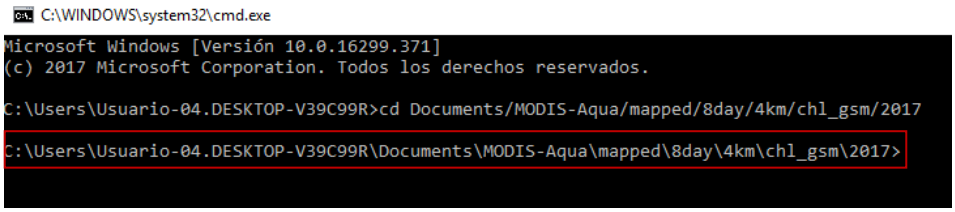


Figura 24. Descripción de acceso a través de consola

Para comenzar con ejecución, se teclea la palabra de entorno reservada *Python*.

Figura 25. Descripción de ejecución para descarga de datos L3 - Palabra reservada Python

Seguidamente, el nombre del script a ejecutar.

Figura 26. Descripción de ejecución para descarga de datos L3 - Nombre del Script

Por último escribe la ruta completa donde reside el archivo de configuración.

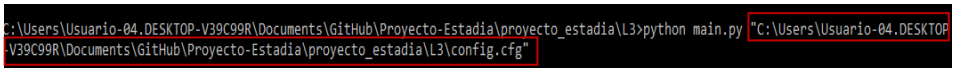


Figura 27. Descripción de ejecución para descarga de datos L3 - Ruta de archivo de configuración

Tras el arranque del script principal, dará inicio a la cadena de procesos que se encargaran de descargar los archivos necesarios acorde a los parámetros definidos.

Figura 28. Ejecución finalizada - Descarga de datos L3

Una vez terminado el proceso de descarga, los datos son almacenados en un directorio raíz (definido por el usuario en su archivo de configuración *ver figura 30*).

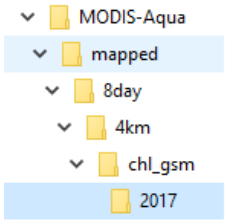


Figura 29. Árbol de directorios creados tras el proceso de descarga

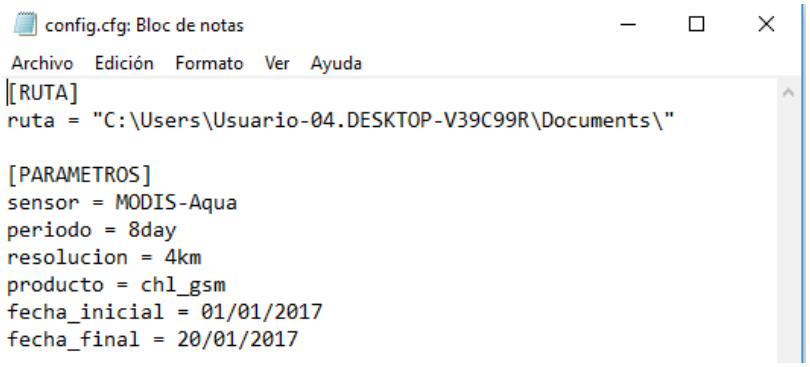


Figura 30. Definición de parámetros de archivo de configuración para la descarga de datos L3

En cuanto a la visualización de los datos descargados, se utiliza la herramienta QGIS que es un Sistema de Información Geográfica de código libre.

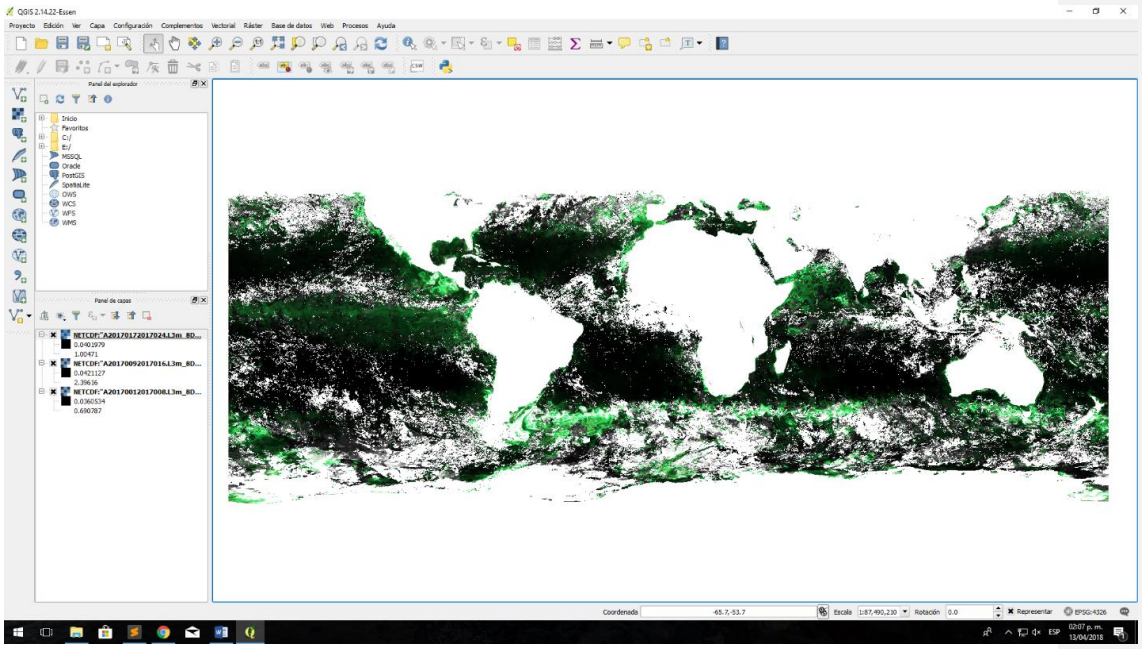


Figura 31. Visualización de datos L3 descargados.

Capítulo 5. Conclusiones

Los objetivos definidos al iniciar el proyecto, fueron logrados con éxito. Por ello, el área de Laboratorio de Percepción Remota Satelital cuenta con un recurso activo para realizar las descargas de datos requeridos de forma automática. A su vez, cada información obtenida mediante este proceso, está debida y claramente organizada en un árbol de directorios correspondiente a los datos. De esta manera la búsqueda de los archivos se vuelve eficiente; el usuario da la carpeta raíz de almacenamiento, por lo que este usuario sabe exactamente donde comenzar a buscar sus datos. Todo fue logrado gracias a la buena comunicación que existió entre el equipo de desarrollo y el cliente, en este caso, el Dr. Juan Pablo Rivera Caicedo investigar de CENIT, se realizó el análisis exhaustivo para obtener los requerimientos que se necesitabas satisfacer con el desarrollo del proceso, de igual forma, se disiparon todas las dudas de carácter técnico (sin incluir la programación) durante el desarrollo del proyecto.

Durante el proceso de desarrollo se dieron a conocer ciertas dificultades técnicas, tales como lo fueron ciertas actualizaciones de la plataforma OCEAN COLOR, éstas impedían la descarga de archivos. Por otra parte, algunos detalles en el funcionamiento de los algoritmos de descargas en la plataforma Linux, sin embargo no presentaron demasiada oposición para concluir con el proyecto. Tras la desarrollo de la cadena de descargas automatizadas de datos satelitales de nivel 2 (L2) y nivel 3 (L3) se dio un retroalimentación y demostración del funcionamiento de la cadena de descargas al Dr. Juan Pablo, responsable del proyecto, quien evaluó el trabajo realizado. Cabe mencionar que a lo largo del proyecto el Dr. Juan Pablo estuvo realizando revisiones sobre avances y haciendo propuestas de mejora por lo cual se realizaron las modificaciones necesarias para el correcto funcionamiento del algoritmo de descarga.

Por parte en la forma al proceso de descarga de los datos se apreció cierta satisfacción, esto porque ahora podrán llevar a cabo esta tarea con más de una archivo a la vez y con las especificaciones detalladas a lo que desean investigar. Mencionar que el Dr. Juan Pablo hizo un énfasis en que los datos que se descargaran deberían estar ordenados en una serie de directorios creados de acuerdo a las características de las especificaciones utilizadas antes de la descarga, esto con la finalidad de la sencillez en el procesado y la búsqueda de los datos que sea necesario. Destacando que esta proceso ahorra tiempo a los investigadores en realizar sus descargas.

El desarrollo del proyecto presento limitaciones en cuestión de tiempo, que anteriormente se mencionó al inicio de este documento, que el tiempo sería un factor nada favorable para la documentación completa del proyecto, esto debido a que su completo desarrollarlo se estimó a un plazo de seis meses. Sin embargo 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 prefirió por solo documentar los avances que estuviesen disponibles hasta la fecha de conclusión de estadías establecida en la agenda que fue puesta a disposición al principio de este proceso, estos detalles fueron comentados con ambos asesores, por parte de los cuales no hubo objeción alguna, estando en acuerdo de que el proyecto se documentara hasta la fecha del 27 de abril del año en curso. Pero ello se debe mencionar que los objetivos establecidos a cuatro meses se cumplieron de forma satisfactoria, los cuales constan de la elaboración de los algoritmos de descarga para los archivos de nivel 2 y nivel 3. Como dato extra, la cadena de descarga de archivos de nivel 3 comenzó a ser utilizada por estudiantes de postgrado antes de la fecha de conclusión estimada y para cuando se terminó el desarrollo del 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 estaba planteado en un inicio, ahora se facilitara a todos los estudiantes de postgrado y practicantes de licenciatura o ingeniería que lo necesiten para realizar proyecto propios o de carrera, 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 mantener cuidado al momento de especificar el apartado de parámetros en el archivo de configuración ya que de los datos que en él se ingresen dependen de los archivos que se desean descargar.

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 Raspberry Pi, etc. Todos con las misma finalidad, ayudar a los investigadores en su trabajo científico.

Igualmente el algoritmo de descargas realizado puede ser extendido, 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.

Referencias

1&1 Digital Guide. (2016, junio 30). El log: el archivo de registro de procesos informáticos. Recuperado el 28 de febrero de 2018, a partir de https://www.1and1.mx/digitalguide/online-marketing/analisis-web/el-log-el-archivo-de-registro-de-procesos-informaticos/

AEM, J. G. B. (2017, noviembre 14). La Percepción Remota. Recuperado el 26 de febrero de 2018, a partir de http://haciaelespacio.aem.gob.mx/revistadigital/articul.php?interior=706

aplicacionesMODIS.pdf. (s/f). Recuperado a partir de http://www.ciga.unam.mx/publicaciones/images/abook_file/aplicacionesMODIS.pdf

CREPAD. (s/f). Recuperado el 27 de febrero de 2018, a partir de http://crepadweb.cec.inta.es/es/plataformas/modis.html

Daniel, R. P., Noela, S. C., Antonio, D. G. J., & Cristina, S. M. P. (2015). *CUESTIONES DE TELEDETECCIÓN*. Editorial UNED.

Documento sin título. (s/f). Recuperado el 28 de febrero de 2018, a partir de http://www.ujaen.es/huesped/pidoceps/telap/perremoc.htm

¿Donde almacenar documentos/fotos/archivos, base de datos o carpeta externa? - Betabeers. (s/f). Recuperado el 28 de febrero de 2018, a partir de https://betabeers.com/forum/donde-almacenar-documentos-fotos-archivos-base-datos-o-carpeta-externa-888/

Guardar imágenes en la base de datos vs guardar imágenes como archivos en el servidor - Stack Overflow en español. (s/f). Recuperado el 28 de febrero de 2018, a partir de https://es.stackoverflow.com/questions/2316/guardar-imágenes-en-la-base-de-datos-vs-guardar-imágenes-como-archivos-en-el-ser

IBM Knowledge Center - Modelo de cliente/servidor. (s/f). Recuperado el 26 de febrero de 2018, a partir de https://www.ibm.com/support/knowledgecenter/es/SSAL2T_9.1.0/com.ibm.cics.tx.doc/concepts/c_clnt_sevr_model.html

Irv. (2005, junio 20). Expresiones regulares. Recuperado el 28 de febrero de 2018, a partir de http://www.desarrolloweb.com/articulos/2033.php

Juan Ardissone. (2011, febrero 9). Métodos GET vs POST del HTTP. Recuperado el 26 de febrero de 2018, a partir de http://blog.micayael.com/2011/02/09/metodos-get-vs-post-del-http/

Mangel. (2018, febrero 14). Peticiones HTTP (GET, POST, PUT, DELETE, ETC). Recuperado el 27 de febrero de 2018, a partir de http://michelletorres.mx/peticiones-http-get-post-put-delete-etc/

Qué es el Web scraping? Introducción y herramientas. (2016, abril 8). Recuperado el 28 de febrero de 2018, a partir de https://sitelabs.es/web-scraping-introduccion-y-herramientas/

Anexos

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 el 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

Si no, si el año actual es igual al año inicial y año actual es diferente al año final:

Realiza la misma instrucción

Si no, si el año actual es mayor al año de inicio y menor al año de fin:

Agrega la url padre al diccionario

Si no, 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