

**Diseño de un entorno de edición y publicación de metadatos por medio de herramientas masivas de importación, basado en la integración de ArcGIS y Geonetwork**

**Autora**

María Fernanda Lozano Bernal

**Tutores**

Laura Sala (LIGIT)

César Martínez Izquierdo (ETC-SIA)

**Universidad Autónoma de Barcelona**

**Departamento de Geografía**

**European Topic Centre- Land Use and Spatial Information**

**Marzo 2013**

## **AGRADECIMIENTOS**

El desarrollo de la memoria como proyecto final de Master en Tecnologías de la Información Geográfica, fue posible gracias a la colaboración y entusiasmo de varias personas y entidades.

Quiero darle las gracias a la Universidad Autónoma de Barcelona, por la realización del Master en Tecnologías de la Información Geográfica.

Al Centro Temático Europeo de Información y Análisis Espacial (ETC-SIA) por permitirme realizar allí el proyecto final de master, al LIGIT por toda su gestión colaboradora. A Miguel Ángel Vargas por la coordinación de las prácticas.

A Dios por permitirme llegar a desarrollar este proyecto final de master.

A mi tutor Cesar Martínez por su paciencia, ayuda y compromiso mostrado durante la realización de mi proyecto.

A mi tutora Laura Sala (LIGIT), un especial agradecimiento por toda su ayuda incondicional, comprensión y entrega no solo durante el proyecto sino durante toda la realización del master.

A mis padres María Teresa, Jorge Enrique, a mi hermanita Ana Teresa, a mi abuelita María Teresa, a Luisita quiero agradecerles porque a pesar de la distancia siempre recibí su ayuda y apoyo incondicional durante toda la realización del Master.

A mis amigos y compañeros del master (MTIG14), gracias por toda su ayuda y por brindarme su amistad.

## TABLA DE CONTENIDO

<b>RESUMEN</b> .....	<b>a</b>
<b>ABSTRACT</b> .....	<b>c</b>
<b>1 INTORDUCCIÓN</b> .....	<b>1</b>
1.1 Centro Temático Europeo de Información y Análisis Espacial (ETC-SIA) .....	1
1.2 Antecedentes.....	5
<b>2. OBJETIVOS</b> .....	<b>6</b>
2.1 Objetivos Generales.....	6
2.2 Objetivos Específicos .....	6
<b>3. MARCO TEÓRICO</b> .....	<b>6</b>
3.1 Que son los Metadatos.....	6
3.1.1 Estándares .....	11
3.1.2 Catálogo de Metadatos .....	17
3.1.3 Transformaciones .....	18
3.1.4 Metadatos en ArcGis .....	19
3.2 Geonetwork .....	26
3.2.1 Búsquedas en Geonetwork .....	29
3.3 Lenguaje Python .....	33
3.3.1 Python en ArcGis .....	37
3.3.2 ArcPy en ArcGis.....	38
<b>4. ANÁLISIS DE REQUERIMIENTOS</b> .....	<b>39</b>
4.1 Requerimientos Funcionales.....	39
4.2 Requerimientos No Funcionales.....	40
4.3 Requerimientos de los Metadatos.....	40
<b>5. METODOLOGÍA</b> .....	<b>42</b>
5.1 Plan de Trabajo.....	42
5.2 Tratamiento de los datos.....	44
5.3 Recolección de la información .....	45
5.3.1 Instalación, personalización y configuración de un catálogo de Metadatos (Geonetwork).....	46
5.3.2 Software Utilizado.....	49
5.3.3 Transformaciones formato XML ISO 19115 a formato ISO 19139 .....	50
<b>6. DESARROLLO DE LA HERRAMIENTA</b> .....	<b>52</b>
6.1 Producción y actualización de metadatos (Metadata Editor) .....	53
6.2 Exportación a XML.....	56
6.3 Metadatos exportados a formato XML ISO 19139 .....	58
6.4 Importación al catálogo de Geonetwork.....	59
6.5 Catálogo de metadatos Geonetwork .....	59
<b>7 CONCLUSIONES</b> .....	<b>73</b>

<b>8 RECOMENDACIONES.....</b>	<b>74</b>
<b>9 BIBLIOGRAFÍA.....</b>	<b>75</b>
<b>10 ANEXOS .....</b>	<b>77</b>

## **LISTA DE FIGURAS**

**Figura 1.** Estados miembros de la EEA

**Figura 2.** Áreas de trabajo de ETC-SIA

**Figura 3.** Distribución de las instituciones en los diferentes países

**Figura 4.** Recursos asociados a información geográfica y ejemplos de metadatos

**Figura 5.** Ejemplos prácticos de elementos a los que se puede aplicar la norma

**Figura 6.** Modelo UML de metadatos

**Figura 7.** Contenido de los metadatos en ArcGis

**Figura 8.** Estilos de los metadatos en ArcGis

**Figura 9.** Almacenamiento de los metadatos en ArcGis

**Figura 10.** Visualización de los metadatos en ArcGis

**Figura 11.** Actualización de los metadatos en ArcGis

**Figura 12.** Interfaces de Geonetwork

**Figura 13.** Búsqueda por defecto en Geonetwork

**Figura 14.** Búsqueda avanzada en Geonetwork

**Figura 15.** Búsqueda por categoría en Geonetwork

**Figura 16.** Búsqueda en la sección de metadatos en el portal Geonetwork

**Figura 17.** Script lenguaje Python

**Figura 18.** Parámetros de entrada y salida B. Secuencia de pasos de generador de metadatos

**Figura 19.** Algunos campos del metadato

**Figura 21.** Descripción metadatos en ArcCatalog

**Figura 22.** Diversos estilos en ArcCatalog

**Figura 23.** Instalación de Geonetwork

**Figura 24.** Portal del catálogo de Geonetwork

**Figura 25.** Sección principal de metadatos en Geonetwork

**Figura 26.** Publicación de los metadatos en Geonetwork

**Figura 27.** Tecnologías empleadas

**Figura 28.** Formato XML 19115

**Figura 29.** Formato XML 19139

**Figura 30.** Script en Python recorre cada una de las carpetas y exporta

**Figura 31.** Producción y generación de metadatos en ArcCatalog

**Figura 32.** Exportación a XML

**Figura 33.** Exportación a formato XML ISO 19139

**Figura 34.** Importación al catálogo de Geonetwork

**Figura 35.** Acceso al portal de Geonetwork

**Figura 36.** Recolección de metadatos (harvesting) en carpetas locales

**Figura 37.** Visualización de adición de metadatos en Geonetwork

**Figura 38.** Numero de metadatos adicionados a Geonetwork

**Figura 39.** Publicación de metadatos en el portal Geonetwork

**Figura 40.** Campo de visualización para insertar la ruta directa a los datos

**Figura 41.** Script para generar la ruta a los datos

**Figura 42.** Visualización de la ruta en el catálogo de Geonetwork

**Figura 43.** Pasos para la configuración de Internet Explorer

**Figura 44.** Pasos para la configuración de Mozilla Firefox

**Figura 45.** Acceso directo a los datos desde Mozilla Firefox

**Figura 46.** Visualización directa de los datos desde Mozilla Firefox

**Figura 47.** Acceso directo a los datos desde Internet Explorer

## **LISTA DE TABLAS**

**Tabla 1.** Clasificación Global de los Estándares de Metadatos (**Paulus Vásquez, C**).

**Tabla 2.** Clasificación ISO de los Estándares de Metadatos (**Brodeur et al.** 2000).

**Tabla 3.** Hojas de estilo en ArcGis (<http://www.resources.arcgis.com>).

**Tabla 4.** Hojas de estilo en ArcGis for Desktop (<http://www.resources.arcgis.com>).

**Tabla 5.** Información básica para publicación y validación de metadatos en Geonetwork

**Tabla 6.** Programación semanal para el desarrollo de las fases.

**Tabla 7.** Diagrama de Gannt



## **RESUMEN**

La presente memoria es el desarrollo del proyecto final del Máster en Tecnologías de la Información Geográfica (MTIG) en su 14ª edición. El Master es realizado en el Laboratorio de Información Geográfica y Teledetección (LIGIT), el que a su vez es organizado por el Departamento de Geografía de la Universidad Autónoma de Barcelona. El presente proyecto es un convenio entre la UAB y el Centro Temático Europeo de Información y Análisis Espacial (ETC-SIA), esta última actúa como entidad receptora de un estudiante del máster para el desarrollo de las sesiones prácticas.

El ETC/SIA (Centro Temático Europeo para la Información y el Análisis Espacial) apoya a la Agencia Europea de Medio Ambiente en el desarrollo de datos espaciales de referencia para el conjunto de Europa. El principal campo de trabajo del centro es el análisis del uso y la cobertura del suelo, lo cual no solo incluye el seguimiento de su evolución temporal sino también el análisis de las consecuencias de los cambios de que pueda ser objeto.

ETC-SIA se ha encargado de ir modificando la aplicación de los metadatos al portal de Geonetwork, con el objetivo no solo de presentar y mejorar la publicación de los mismos sino permitiéndole a los posibles diferentes usuarios interesados en el intercambio de información cartográfica.

Existe una utilización de gran cantidad de información geográfica en el marco del ETC-SIA y del resto de los proyectos del grupo Interfase, y debido a esto se ha presentado la necesidad de catalogar y documentar correctamente la información, para esto se necesitaba crear un sistema que permitiera facilitar la búsqueda y el uso de la información existente en el Centro, y a su vez mantener las bases de datos actualizadas de la manera más cómoda posible.

El proyecto final de máster propuesto por la entidad y titulado como “Diseño de un entorno de edición y publicación de metadatos por medio de herramientas masivas de importación, basado en la integración de ArcGIS y Geonetwork”, propuso la actualización del catálogo

de información geográfica de la ETC-SIA, por otro lado la instalación, personalización y configuración de un catálogo de metadatos (Geonetwork), se elaboró igualmente, una herramienta para la búsqueda de metadatos en formato ESRI XML, posteriormente se necesitaba realizar una transformación a formato XML ISO 19139 ya que Geonetwork no acepta otro formato y finalmente con la realización de la configuración o modificación de Geonetwork se obtuvo que los diferentes resultados logrados a partir de una búsqueda dieran como resultado la ruta a los datos.

Para el desarrollo del proyecto se tuvo en cuenta cuatro fases diferentes las cuales fueron el análisis, diseño del software, desarrollo e implementación. En cuanto al desarrollo e implementación es importante resaltar que lo hecho anteriormente fue realizado por medio de un script en lenguaje Python, se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y en menor medida, programación funcional. Finalmente el script se ejecutó para todos los metadatos que se encontraban dentro del ETC-SIA.

Dentro de la memoria se reflejan las ventajas que esto conlleva como lo es la rapidez y la eficacia que obtienen los usuarios al hacer uso del entorno diseñado con el fin de que los resultados obtenidos de una búsqueda contengan la ruta directa a los datos. No obstante cada vez se puede ir realizando mejoras dentro del entorno diseñado para que este a su vez sea óptimo.

**Palabras Claves:** Geonetwork, ArcGis, Metadatos, Python, ETC-SIA.

## **ABSTRACT**

This report is the final project of Master of Geographic Information Technologies (mtig) in its 14th edition. The Master is conducted at the Laboratory of Stratigraphy (LIGIT), which in turn is organized by the Department of Geography at the Autonomous University of Barcelona. This project is a partnership between UAB and the European Topic Centre on Spatial Information and Analysis (ETC-SIA), the latter acts as a student organization receiving the master for the development of the practical sessions.

The ETC / SIA (European Topic Centre for Information and Spatial Analysis) supports the European Environment Agency in developing spatial reference data for the whole of Europe. The main field of work is the analysis center use and land cover, which not only includes monitoring their temporal evolution but also the analysis of the consequences of the changes that can be.

ETC-SIA has undertaken gradually modify the application of metadata to GeoNetwork portal, aiming not only to present and improve the publication of the same but allowing different potential users interested in mapping information exchange.

There is a lot of use of geographic information within the ETC-SIA and the rest of the interface group projects, and because of this has presented the need to label and properly document information, it was necessary to create a system that would facilitate the search and use the information in the center, and in turn keep the databases updated as comfortable as possible.

The final master project proposed by the entity and entitled as "Design of an editing environment and publishing metadata through mass import tools, based on the integration of ArcGIS and Geonetwork" proposed updating the geographic information catalog ETC-SIA, second installation, customization and configuration metadata catalog (Geonetwork) was prepared similarly, a tool for searching XML metadata in ESRI format then needed a transformation to ISO 19139 XML format as Geonetwork not accept another format and finally to the embodiment or modification of the configuration of Geonetwork was obtained that different results from a search would result in the data path.

For the project took into account four different phases which included analysis, software design, development and implementation. On the development and implementation is important to note that what came before was done by a Python script language, is a multi-paradigm programming language, since it supports object oriented, imperative programming and to a lesser extent, functional programming. Finally the script is executed for all metadata were within the ETC-SIA.

Within the memory reflects this entails advantages such as the speed and efficiency that users get the intended use environment to which the search results of a direct route to contain the data. But every time you can go to make improvements within the environment designed for this in turn is more optimal.

**Keywords:** GeoNetwork, ArcGIS, Metadata, Python, ETC-SIA.

## **1 INTORDUCCIÓN**

La presente memoria contiene cada uno de los resultados del proyecto final de Master de Tecnologías de la información Geográfica de la décimo cuarta edición, organizado por el Departamento de Geografía de la Universidad Autónoma de Barcelona e impartido por el laboratorio de Información Geográfica y Teledetección (LIGIT), proyecto que tuvo una duración de tres meses prácticos, posterior a la finalización de las practicas se procedió a la redacción de la memoria, el desarrollo de esta se hizo con colaboración con el Centro Temático Europeo de Información y Análisis Espacial (ETC-SIA), el cual consistió en el Diseño de un entorno de edición y publicación de metadatos por medio de herramientas masivas de importación, basado en la integración de ArcGIS y Geonetwork

### **1.1 Centro Temático Europeo de Información y Análisis Espacial (ETC-SIA)**

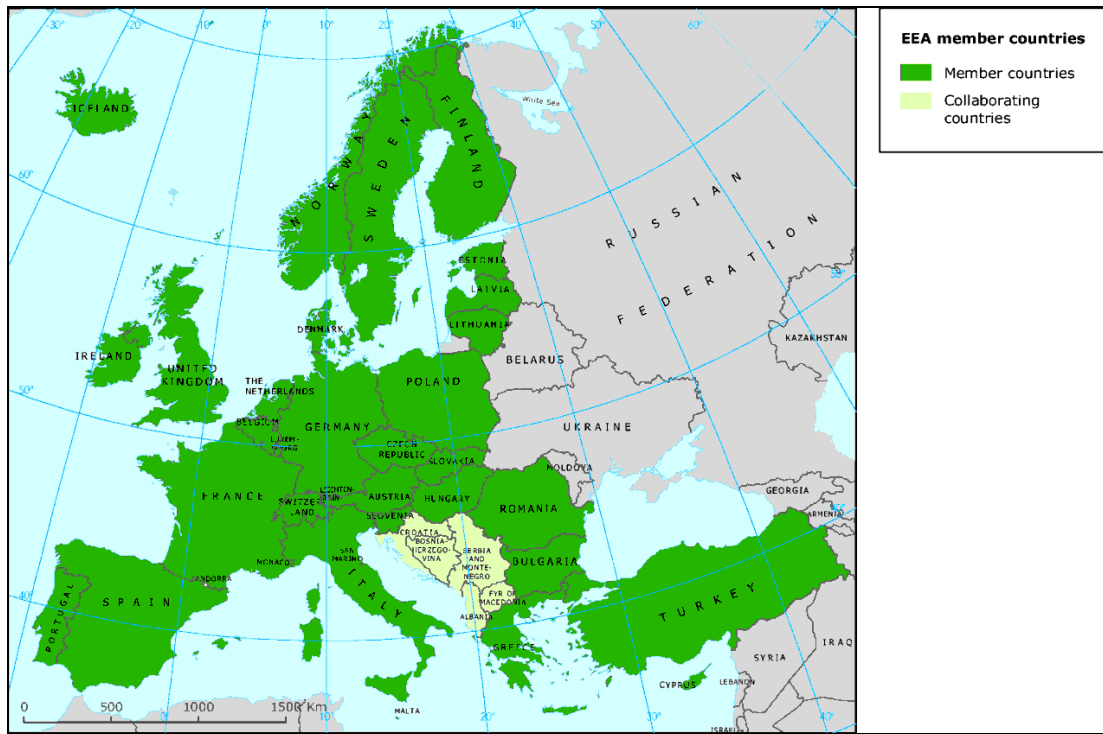
El ETC-SIA, es el encargado de apoyar a la Agencia Europea de Medio Ambiente en el desarrollo de datos espaciales para el conjunto de Europa, el principal trabajo es el análisis del uso y la cobertura del suelo, es decir no solo incluye el seguimiento de su evolución temporal sino también el análisis de las consecuencias de dichos cambios (**ETC-SIA 2010**).

La Agencia Europea de Medio Ambiente fue creada en el año de 1994. La AEMA tiene como objetivo principal dar soporte al desarrollo sostenible y ayudar a alcanzar mejoras significativas y medibles en el medio ambiente europeo mediante la provisión de información actualizada, concreta, relevante y fiel a los agentes políticos y al público en general (**ETC-SIA 2010**).

Las tareas principales de la AEMA:

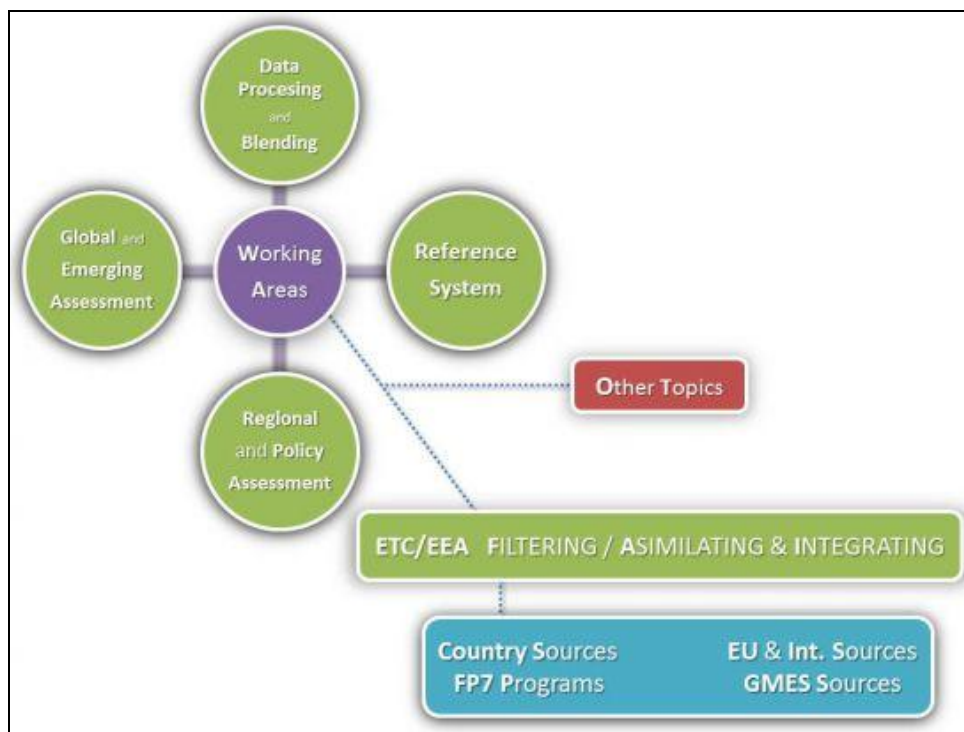
- Creación de redes de información → EIONET.
- Elaboración de informes.
- Facilitar el acceso a la información ambiental.

Los estados miembros AEMA se pueden visualizar en el siguiente mapa.



**Figura 1.** Estados miembros de la EEA

Por otro lado el trabajo del ETC-SIA se estructura en cuatro áreas como se ve en la figura a continuación (**ETC-SIA 2010**).



**Figura 2.** Áreas de trabajo de ETC-SIA

El ETC-SIA, ofrece datos en series temporales, evalúan la eficacia de los instrumentos políticos y contribuyen a los diversos informes que elabora la Agencia. Por otra parte se basa en la gestión y control de calidad de bases de datos geográficos y en la evaluación ambiental (**ETC-SIA 2010**).

Una parte importante del trabajo que se realiza en el centro está dedicado a recopilar, gestionar, analizar y mostrar datos espaciales de usos del suelo. El ETC/SIA está involucrado en el desarrollo de soluciones de código abierto para cumplir tanto con la Directiva INSPIRE como con la política europea SEIS (Shared Environmental Information System), formando parte de una red de recursos distribuidos y simplificando el acceso del público a los datos (**ETC-SIA 2010**).

El ETC es parte de **EIONET** (European Environment Information and Observation Network) y a su vez coopera con otras instituciones europeas como el **JRC** (Joint Reserach Centre), **Eurostat**, las distintas Direcciones Generales de la Comisión Europea y **ESPON**(European Observation Network for Territorial Development and Cohesion) (**ETC-SIA 2010**).

Los representantes de universidades, autoridades nacionales, instituciones gubernamentales y empresas privadas son las principales colaboradoras con el fin de proporcionar la información territorial más actualizada y los resultados analíticos más precisos para el público europeo. Además, están vinculados a convenios regionales y a organizaciones no gubernamentales internacionales que apoyan el desarrollo y aplicación de los datos generados en sus sistemas de datos. Los socios pertenecen a la comunidad científica internacional y también se encuentran involucrados en proyectos de origen investigativo a nivel nacional e internacional (**ETC-SIA** 2010).

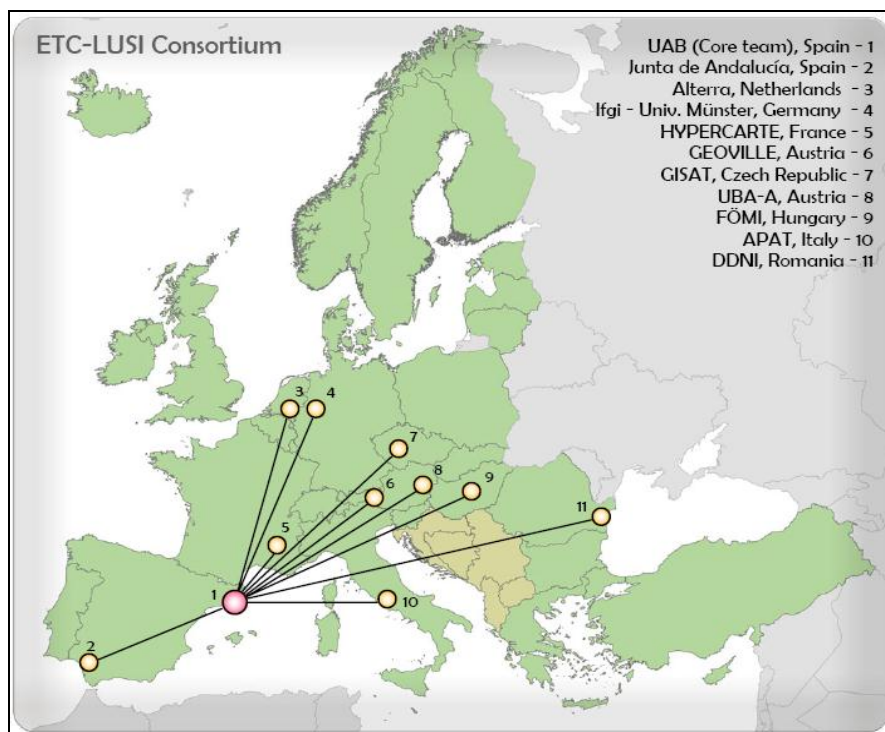
Algunos de los proyectos recientes en el Centro son:

- FP7 - PEGASO, People for Ecosystem-based Governance in Assessing Sustainable development of Ocean and coast
- ESPON – EU-LUPA, European Land Use Patterns
- GMES/FP7 – MS. MONINA, Multi-scale Service for Monitoring NATURA2000 Habitats of European Community Interest
- GMES/FP7 – HELM, Harmonised European Land Monitoring
- ESPON – ESaTDOR, European Seas and Territorial Development Opportunities and Risks

Los Centros Temáticos son consorcios de organizaciones de expertos en una temática determinada, contratados por la EEA para ayudarla a ejecutar su plan de trabajo.

El Centre Temàtic de Informació Espacial y Análisis (<http://www.etc-sia.eionet.europa.eu>), consorcio liderado per la UMA/UAB. Cuenta con 10 instituciones europeas de 9 países diferentes.





**Figura 3.** Distribución de las instituciones en los diferentes países

## 1.2 Antecedentes

Debido a la unión existente entre el ETC-SIA y el LIGIT, se han venido desarrollando años atrás proyectos, algunos de estos relacionados con el campo de los metadatos y con los diferentes entornos relacionados con los sistemas de información geográficos. Se desarrolló con anterioridad un proyecto titulado “Millora I Implementació de l’eina open-source AC2Geonetwork per ArcCatalog”, cuyo autor es Óliver Sánchez Camacho García (2011). El cual consiste en generar nuevos módulos de publicación de los metadatos que adoptan los nuevos formatos existentes así como la automatización del proceso para grandes volúmenes de datos, ya que inicialmente la herramienta tan solo permitía la publicación de archivos cartográficos en formato shape (.shp.)

El desempeño de esta demanda requería crear nuevos procesos en visual basic (VB), que integren dentro de la extensión de ArcCatalog y publique al portal web Geonetwork en forma requerida y establecida.

## **2. OBJETIVOS**

El objetivo principal del proyecto es el Diseño de un entorno para la edición y publicación de metadatos, mediante herramientas masivas de importación, basado en la integración de ArcGis y Geonetwork.

### **2.1 Objetivos Generales**

- Facilitar la búsqueda y el uso de la información existente en el centro.
- Mantener las bases de datos actualizadas de la manera más cómoda posible.

### **2.2 Objetivos Específicos**

- Instalar, personalizar y configurar un catálogo de metadatos (Geonetwork)
- Preparar una herramienta para la búsqueda de metadatos en formatos ESRI XML y transformación a formato XML ISO 19139.
- Configurar o modificar Geonetwork de forma que los resultados de una búsqueda específica contengan la ruta a los datos.

## **3. MARCO TEÓRICO**

En este ítem se explicara detalladamente cada uno de los conceptos que fueron tenidos en cuenta durante el desarrollo del proyecto lo mismo se aplica al lenguaje que fue utilizado para el desarrollo del script.

### **3.1 Que son los Metadatos**

Los metadatos, en sí, no suponen algo completamente nuevo dentro del mundo bibliotecario. Según Howe (1993), el término fue acuñado por Jack Myers en la década de los 60 para describir conjuntos de datos. La primera acepción que se le dio (y actualmente la más extendida) fue la de dato sobre el dato, ya que proporcionaban la información mínima necesaria para identificar un recurso (**Caplan**, 1995).

Otros autores amplían el concepto de “dato sobre el dato”, afirmando que los metadatos también incluyen información sobre el contexto, contenido y control. De este modo, se

alcanzan objetivos como describir, identificar y definir un recurso para recuperar, evaluar y auto identificar (**Caplan**, 1995).

A su vez los metadatos pueden describir colecciones de objetos y también los procesos en los que están involucrados, describiendo cada uno de los eventos, sus componentes y cada una de las restricciones que se les aplican. Los primeros metadatos (digitales) y sus bases se desarrollan a finales del siglo XX, cuando emergen múltiples estándares de codificación, lenguajes y protocolos que se utilizan en la generación y uso de catálogos (**Paulus Vásquez**, C)

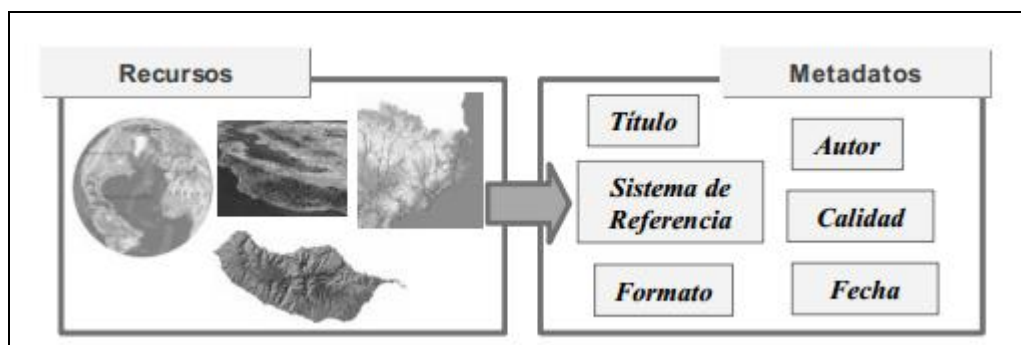
El concepto de metadatos en el campo de la Información Geográfica, fue aplicado en la década de los 90, donde se produjo una migración conceptual desde una primera concepción de los metadatos como “datos acerca de los datos”, tal y como los define ISO19115, hasta la idea de metadatos como datos que describen no sólo datos sino también servicios (“datos acerca de datos y servicios”), como pueden ser los servicios de publicación de mapas en Internet, servicios de transformación de coordenadas, servicios de nomenclátor, etc (**Pascual -Rodríguez A**, *et al* ).

Los datos geográficos digitales pretenden modelar y describir el mundo real para su posterior análisis y visualización mediante medios muy diversos. Con el fin de definir una estructura que sirva para describir los datos geográficos se creó la norma Internacional ISO 19115:2003- Geographic Information Metadata (**Ballari**, *et al.* 2008).

Para la elaboración de esta norma fue necesaria la colaboración de 33 países miembros de ISO/TC211 y un total de 16 países que aportaron expertos al Grupo de Trabajo (WG) encargado de su definición. En el año 2003 se aprobó el texto definitivo como Norma Internacional de metadatos que fue adoptada como Norma Europea por CEN/TC 287 en 2005. AENOR (Asociación Española de Normalización) ha decidido también su adopción como Norma Española, con la identificación: UNE-EN ISO19115 (**Ballari**, *et al.* 2008).

El concepto de metadatos se está convirtiendo en una herramienta familiar para aquellas personas que trabajan con información espacial. Así la leyenda que encontramos en un mapa es un ejemplo de aplicación de los metadatos que nos proporciona información sobre

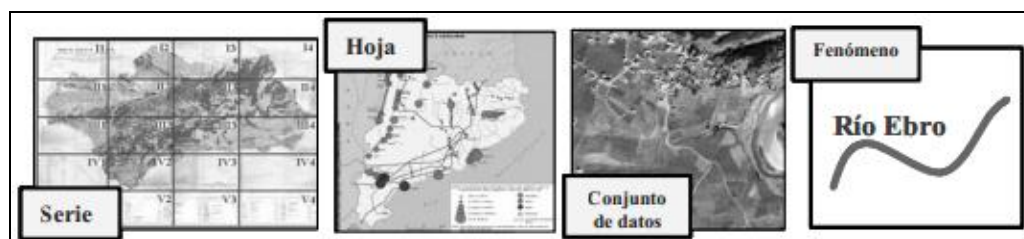
el autor, la fecha de publicación, la escala y otras características propias del mapa (Figura 4).



**Figura 4.** Recursos asociados a información geográfica y ejemplos de metadatos

Mediante la definición de elementos de metadatos se va a poder describir información sobre la identificación, la extensión, la calidad, el modelo espacial, temporal, la referencia espacial y la distribución de los datos geográficos. Esta norma se aplica a: (Ballari, *et al.* 2008).

- La catalogación de conjuntos de datos y la descripción completa de conjuntos de datos.
- Diferentes niveles de información como conjuntos de datos geográficos, series de conjunto de datos, fenómenos geográficos individuales, propiedades de los fenómenos, etc. (Figura 5)



**Figura 5.** Ejemplos prácticos de elementos a los que se puede aplicar la norma

Esta Norma de Metadatos es de una gran complejidad e incluye una extensa serie de elementos de metadatos, unos obligatorios y otros opcionales. El documento de aprobación

de esta norma consta de 140 páginas, 409 ítems y 27 listas controladas mediante las cuales son definidos los posibles valores validos de ciertos campos (Ballari, *et al.* 2008).

En la Norma ISO 19115, los metadatos para datos geográficos son representados mediante paquetes UML (Unified Modelling Language). Cada uno de los paquetes UML contiene una o más entidades (clases UML), que pueden estar especificadas (*subclase*) o generalizadas (*superclase*). Las entidades contienen elementos (atributos de clases UML) que identifican las unidades o ítems discretos de metadatos (Pascual R. A, 2003).

En la figura se puede ver la disposición de los paquetes. Los metadatos están completamente especificados en los diagramas de modelado UML y el diccionario de datos para cada paquete.

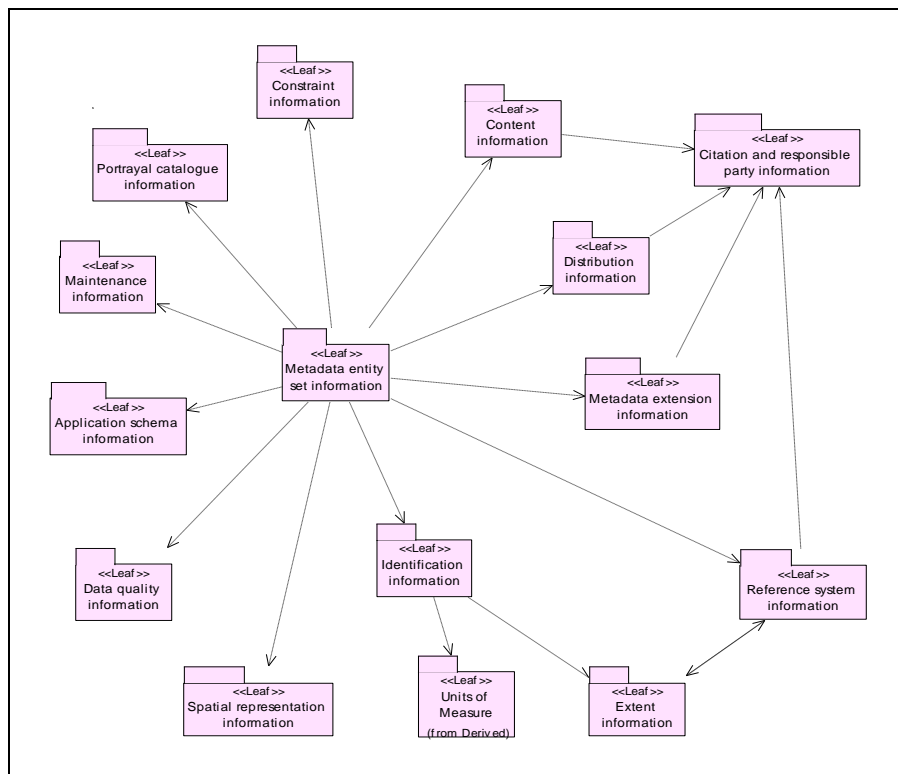


Figura 6. Modelo UML de metadatos

Posterior a la norma ISO 19115 se realizó la versión definitiva de la norma ISO19139 la cual fue publicada el 17 de abril de 2007, en la página web de la ISO, allí es posible realizar la descarga del conjunto de ficheros siguiendo la especificación del W3C para el esquema de XML. Esta es la implementación del esquema necesario para validar la apariencia de los ficheros de metadatos de información geográfica. Su aplicación definitiva permitirá unificar la apariencia de los ficheros XML y con ello impulsar la interoperabilidad entre todas las fuentes de información (**Amaro, A. 2007**).

La creación del esquema XML para la validación de los metadatos de información geográfica obtenidos en la norma ISO 19115, se llevó a cabo por el grupo EDEN del IGN en Francia, durante un proceso de varios años (**Amaro, A. 2007**).

En la norma ISO19139 de implementación del esquema se recogen las reglas para generar un esquema XML el cual permita validar ficheros XML con metadatos de la norma ISO 19115; el objetivo de crear esta norma es ajustar el amplio rango de posibilidades que se ofrecen para plasmar en un medio digital los metadatos de información geográfica. Por lo tanto, la norma ISO19139, como se ha mostrado es un medio común la descripción, validación e intercambio de metadatos relacionados con conjuntos de datos geográficos, series y atributos. (**Amaro, A. 2007**).

En ArcGis, los metadatos son aquellos que proporcionan la descripción de cada elemento que se encuentra dentro de ArcGis, en los metadatos de un elemento se puede registrar la información necesaria sobre este elemento. Dentro de los metadatos se puede incluir información precisa y reciente, las restricciones asociadas, los procesos importantes en el ciclo de vida como generalizar las entidades.

Finalmente los metadatos permiten generar distintos puntos de vista conceptuales para sus usuarios o sistemas; y liberan de tener conocimientos avanzados sobre la existencia o características del objeto que describen. Estas también permiten el intercambio de información, a su vez muestran una descripción precisa y discreta de los recursos.

### 3.1.1 Estándares

Los estándares para metadatos son considerados como una forma genérica de organizar los metadatos, los estándares de los metadatos varían de uno a otro básicamente en la información que considera importante es decir aquella que consideran como un metadato.

Existe una cantidad significativa de datos recopilados y desarrollados por la comunidad geoespacial, y frecuentemente esos datos son útiles para muchas aplicaciones. Pero debido a la falta de medios y mecanismos para discernir qué datos existen, que errores tienen y como acceder y usar los mismos evitando una inútil duplicación de esfuerzos se hace necesario alguna solución (**Berlanga, A; et al.**).

Para facilitar la comprensión global de los metadatos es necesario clasificarlos en tres grandes grupos los cuales se muestran en la tabla a continuación.

**Tabla 1.** Clasificación Global de los Estándares de Metadatos (**Paulus Vásquez, C.**)

Tipo	Definición	Ejemplos
Administrativos	Metadatos utilizados para el manejo y administración de los recursos de información	<ul style="list-style-type: none"><li>• Control de versión</li><li>• Reutilización de contenidos</li></ul>
Descriptivos y Descubrimiento	Metadatos utilizados para describir, descubrir o identificar recursos de información	<ul style="list-style-type: none"><li>• Índices especializados</li><li>• Taxonomías</li></ul>
Técnicos, modelos	Metadatos relacionados con cómo funciona un sistema o de cómo se interrelacionan sus componentes	<ul style="list-style-type: none"><li>• Estructura de formatos digitales</li><li>• Modelos de intercambio de Metadatos</li></ul>

El estándar administrativo hace referencia sobre cuando y como fue creado un objeto, quien es el responsable, el acceso o registrar su contenido. El segundo tipo de estándar es

conocido como descriptivos y de descubrimiento se refieren a la información provista para encontrar, describir y distinguir cada uno de los objetos de información. El último tipo de estándares son conocidos como técnicos y modelos, en cuanto a los técnicos se refiere a los estándares de metadatos relacionados con los elementos que describen cómo funciona un sistema o cómo debe ser interpretado. Mientras los de tipo modelo, tienen relación con las piezas de un objeto de información compuesto, en términos de cómo se interrelacionan cada uno de sus componentes, un ejemplo sencillo en el contexto de un libro, llegaremos a un tema deseado si seguimos el número de página indicado en el índice y que además las páginas están ordenadas (**Paulus Vásquez, C**).

Sin embargo hay que tener en cuenta que los límites entre las diferentes categorías tienden a ser difusos, ya que muchos de los metadatos no caben en una sola de las categorías.

A continuación se describirán estándares de metadatos, de metadatos geográficos, y estándares para transferencia de datos.

### **Estándares de Metadatos:**

➤ **Estándares World Wide Web Consortium (W3C):** Ha desarrollado un interés en los metadatos RDF *“(modelo de datos y sintaxis para los metadatos de los recursos web) y la especificaciones para la Plataforma para la Selección de Contenido en Internet (PICS)”*. RDF permite la codificación, intercambio y reutilización de metadatos estructurados. Inicialmente fue diseñado para permitirle a los padres el poder controlar el acceso de los niños a ciertos sitios en internet (<http://www.w3c.org/>).

➤ **Estándar de Metadato Dublin Core:** Este es un estándar que permite facilitar la recuperación de objetos tales como documentos, en un ambiente de red como puede ser por internet (**DCE, 1998**).

Dublin Core se concentra en la descripción de las propiedades intrínsecas tales como el contenido intelectual (título, autor o fuente) o forma física (formato).

Dublin Core contiene 15 elementos de metadatos los cuales se dividen en tres grupos.



- Contenido: Título, tema, descripción, fuente, lenguaje, cobertura.
  - Propiedad Intelectual: Autor, editor, colaborador.
  - Instanciación: Fecha, formato, tipo, identificador.
- **Estándares de Metadatos Geográficos:** Fue en junio de 1992, cuando el Comité Federal de Datos Geográficos (FGDC) de Estados Unidos, auspicio un “Foro de intercambio de información de datos espaciales”. Después de algunas revisiones preliminares en los años 1992 y 1993, se aprobó la primera versión estándar de Contenido de Metadatos Geoespaciales (CSDGM) en junio de 1994. Poco después la Organización Internacional de Estándares (ISO), realizo actividades similares a través de su comité técnico 211 (ISO/TC 211) (<http://www.fgdc.gov>).

La Organización Internacional de Estándares (ISO) fue creada en 1947. Conformada por capítulos de 143 países. El objetivo principal de este Comité es facilitar el intercambio internacional de bienes y servicios así como desarrollar cooperación entre actividades intelectuales, científicas, económicas y tecnológicas (<http://www.iso.org>).

### **Metadatos Geográficos:**

En cuanto al área de información geográfica la serie ISO 19100 estandariza los aspectos de mayor importancia en cuanto a la descripción y administración de la información y los servicios geográficos. El capítulo ISO 211 (ISO/TC 211), establece en sus estándares una manera para describir objetos en un contexto geográfico cubriendo su ubicación, geometría y su estructura, sus propiedades y también sus metadatos. Este estándar proporciona primitivas geométricas y técnicas que permiten adecuarse a estas propiedades las cuales definen a los objetos geográficos como entidades del mundo real (<http://www.iso.org>).

En la siguiente tabla se mostrara una clasificación con una breve descripción de los estándares ISO de metadatos.

**Tabla 2.** Clasificación ISO de los Estándares de Metadatos (**Brodeur et al.** 2000).

ISO	19103	Lenguaje del esquema conceptual	Selección del lenguaje del esquema conceptual que cubra las necesidades de los modelos información geográfica.
ISO	19107	Esquema espacial	Define conjunto estándar de tipo de datos espaciales y operaciones para espacios geométricos y topológicos.
ISO	19105	Conformidad y pruebas	
ISO	19108	Esquema temporal	Define características temporales y funciones que se desarrollan en un contexto geoespacial.
ISO	19111	Referencia espacial por coordenadas	Define metodología para documentar el sistema coordinado de referencia que apoya la descripción de una posición.
ISO	19115	Metadatos	Define la estructura y el contenido de los componentes de metadatos que describen el conjunto de datos.
ISO/TR	19120	Estándares funcionales	Reporte Técnico
ISO/TR	19121	Imágenes y datos cuadrículados	Reporte Técnico
ISO	19139	Esquema implementación XML (metadatos)	Especificación técnica que desarrolla una implementación en XML del modelo de metadatos

			descrito por ISO 19115.
--	--	--	-------------------------

Varios estándares de metadatos geospaciales son producidos por comités ISO incluyendo ISO 19115 información geográfica-metadatos e ISO 19119 información geográfica-servicios los cuales son estándares de contenido, también hay estándar como ISO 19139 información geográfica-metadatos-implementación del esquema XML, el cual proporcionan un esquema y define un formato XML, en el que se almacena el contenido de los metadatos ISO 19115 e ISO 19119, por lo que se procedió a la implementación del estándar ISO 19139.

Open Gis Consortium (OGC), organización sin ánimo de lucro fundado en 1994, la cual se enfoca primordialmente en desarrollar especificaciones tales como ISO e implementarlas OpenGis plantea un catálogo de datos geospaciales (catalog services), el cual está pensado para ser distribuido a través de la red el cual permita la ejecución de búsquedas basándose en el lenguaje estándar de consulta y la presentación de resultados mediante perfiles especiales de XML. Esto ha permitido facilitar la búsqueda de diferentes trabajos donde cada uno de ellos puede contar con información propia y acceder a los otros de manera sencilla (<http://www.opengis.org>).

Otro estándar contemplado entre los metadatos geográficos es el Content Standard for Digital Geospatial Metadata (CSDGM), este estándar fue desarrollado por el Comité Federal de Datos Geográficos de los Estados Unidos, el objetivo principal de este estándar era proporcionar un conjunto común de terminología y definiciones de los datos geospaciales digitales (<http://www.fgdc.gov>).

Dicho estándar define la información que requiere el usuario para determinar la disponibilidad de un conjunto de datos geospaciales y así mismo sabes si estos datos son apropiados para el uso que se les da. De igual manera muestra cómo se puede acceder a ellos y transferirlos en forma exitosa (<http://www.fgdc.gov>).

## Estándares para Transferencia de Datos:

- **SGML (Standard Generalized Markup Language):** Es una técnica para la representación de documentos en formato electrónico. Fue aprobado como estándar internacional ISO 8879 en 1986. Fue diseñado para satisfacer la necesidad de compartir datos por medio de documentos independientes a la aplicación. Utiliza una definición de tipo de documento DTD (Document Type Definition) lo cual lo utiliza para determinar la estructura del documento. SGML es un lenguaje potente y versátil sin embargo estas propiedades lo hace un lenguaje difícil de utilizar y inadecuado para el desarrollo de la web (STU, 2000).
- **HTML (Hypertext Markup Language):** Lenguaje sencillo que promueve el uso del internet. Herramienta, protocolo o lenguaje que permite la integración de datos en un formato que pueda validarse de forma automática para verificar su corrección sintáctica y una estructura adecuada de su contenido.
- **XML (Extensible Markup Language):** En 1996 W3C desarrollo un nuevo lenguaje de marcado estándar XML, este formato permite la transferencia y archivo de documentos de texto. Al igual que HTML delimita su contenido por medio de etiquetas, aunque existen diferencias significativas entre XML y HTML. Se han desarrollado una serie de formatos basados en XML, los más interesantes son los enfocados en el área grafica como SVG, X3D este último es para el desarrollo de aplicaciones en tercera dimensión (<http://www.w3.org/TR/WD/-xml-lang#sec1.1>).

Finalmente los estándares en ArcGis representan un documento que identifica el contenido el cual describe recursos geospaciales como mapas, servicios de mapas, datos vectoriales, imágenes e incluso tablas y herramientas relevantes a su trabajo espacial. Un estándar de metadatos también puede proporcionar un esquema XML, el cual demuestra el formato en

el cual se almacena el contenido. Por lo general los estándares se ratifican mediante conjunto de estándares nacionales e internacionales (<http://www.help.arcgis.com>).

Un perfil de metadatos, es un documento el cual adapta un estándar de metadatos. Un perfil reduce la cantidad total de elementos de metadatos definidos por un estándar, a su vez un perfil puede restringir la opcionalidad de un elemento de metadatos, logrando hacerla obligatoria mientras antes era opcional; aunque un perfil no puede convertir elementos obligatorios en opcionales. Un ejemplo de un perfil de metadatos es el perfil norteamericano de ISO 19115:2003 – información geográfica-metadatos que se creó y adopto en conjunto por los Estados Unidos y Canadá.

### **3.1.2 Catálogo de Metadatos**

Los registros de metadatos se suelen publicar a través de sistemas de catálogos, en ocasiones también denominados directorios o registros. Los catálogos electrónicos no difieren demasiado de los catálogos tradicionales de una biblioteca excepto por el hecho de ofrecer una interfaz estandarizada de servicios de búsqueda (**Ballari, et al.** 2008).

Un Catálogo de metadatos, es una colección de metadatos que describen elementos los cuales están disponibles en una amplia variedad de fuentes. Normalmente los catálogos de metadatos requieren que los metadatos sigan un estándar o un perfil de metadatos, ya que la información que se encuentra estandarizada facilita a las personas de comunidades distintas, industrias y países, compartir información y que a su vez se comprendan. Por otro lado los catálogos de metadatos también pueden sugerir o requerir información adicional que no es obligatoria en un estándar o perfil; estas opciones de búsqueda solo funcionan si la información que relacionan existe en los metadatos. Si se desea publicar metadatos en un catálogo de metadatos hay que asegurarse de cumplir con cada uno de los requisitos del catálogo, además de aquellos de un estándar de contenido o perfil cuando edite los metadatos (**Ballari, et al.** 2008).

### 3.1.3 Transformaciones

Una transformación es una conversión unívoca de una lista de elementos en otra. Una transformación permite que un destino identifique una asignación directa entre sus entradas y salidas. Las transformaciones no son arbitrarias, pero están limitadas por una sintaxis especial.

Actualmente uno de los principales problemas a los que se enfrenta la generación de metadatos ISO 19115 es la dificultad para intercambiar metadatos creados con distintas herramientas. Esto es debido a la inexistencia de una estructura de almacenamiento estándar. La Internacional Organization for Standardization “ISO” está tratando salvar este inconveniente a través del desarrollo de la Especificación Técnicas 19139 “*Geographic Information –Metadata- XML schema implementation*”, que proporciona un mecanismo para volcar el contenido de los metadatos definidos de acuerdo a ISO19115 en XML

ArcCatalog actualmente utiliza una versión desfasada del estándar ISO19115, lo que significaría que los metadatos creados con dicha herramienta no serán entendidos por otras herramientas con la versión actualizada del estándar.

Dentro de ArcCatalog se proporciona un nuevo traductor de metadatos con la versión actual de ArcGIS for Desktop: el traductor de ArcGIS a ISO 19139, ARCGIS2ISO19139.xml. Asimismo, cualquier modelo de geoprocresamiento o secuencia de comandos de Python que exporta metadatos al formato ISO 19139 se debe actualizar para utilizar este traductor ([http:// www.resources.arcgis.com](http://www.resources.arcgis.com)).

La versión anterior del traductor de ArcGIS o Esri-ISO a ISO 19139, ESRI\_ISO2ISO19139.xml, se sigue proporcionando; sin embargo, se han encontrado varios problemas conocidos con este traductor que no se pueden resolver. Los metadatos 9.3.1 existentes en el formato ESRI-ISO se deben actualizar al formato de metadatos de ArcGIS para poder exportar correctamente el contenido de los metadatos del

elemento al formato ISO 19139 con el traductor de ArcGIS a ISO 19139, ARCGIS2ISO19139.Xml ([http:// www.resources.arcgis.com](http://www.resources.arcgis.com)).

La transformación de la ISO 19115 y afines ISO / TC 211 abstracto UML en modelos de esquemas XML. Para propósitos informativos, secuencias de comandos para realizar la transformación se hace referencia ([http:// www.resources.arcgis.com](http://www.resources.arcgis.com)).

En Geonetwork el perfil de metadatos ISO 19139 usado para describir datos geográficos y servicios se basa en el estándar 19115:2003 y proporciona información relacionada con la identificación, mantenimiento y restricciones, la extensión temporal y espacial, la representación espacial y referencia, la calidad y distribución de un juego de datos geográficos (<http://geonetwork-opensource.org/>).

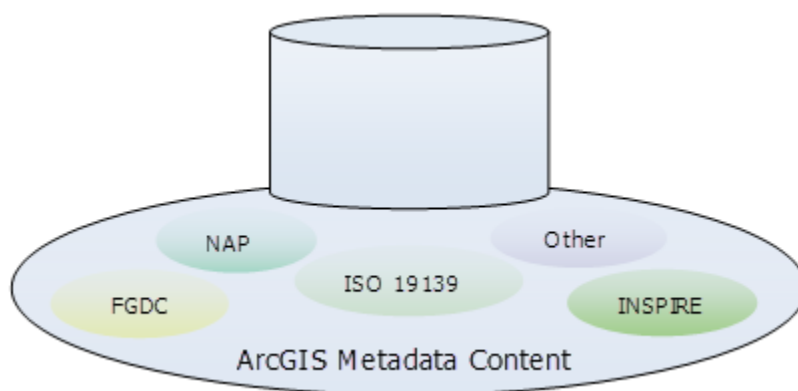
#### **3.1.4 Metadatos en ArcGis**

Como se había mencionado anteriormente cuando se hizo referencia a qué eran los metadatos; encontramos que en ArcGis todo elemento tiene una descripción a la que se denomina metadatos.

En los metadatos de un elemento se puede registrar la información importante que cualquier organización conozca sobre ese elemento. Dentro de ArcGis para la visualización y edición de los metadatos se realiza en la pestaña de Descripción.

Arcgis proporciona acceso a los metadatos para todos los elementos como un documento XML. El encargado de leer y almacenar la información en un conjunto específico de elementos XML es el editor de metadatos. Conjuntamente esos elementos son conocidos como el formato de metadatos de ArcGis. El formato de metadatos consta de elementos que pueden almacenar el contenido de los estándares de metadatos que admite ArcGis Desktop. Incluyendo el contenido de metadatos de FGDC, el contenido de metadatos de ISO 19139, el contenido de metadatos del Perfil norteamericano y todo el contenido de metadatos INSPIRE. También incluye el contenido interno de esta información es útil, pero no se

incluye en los perfiles y estándares de metadatos oficiales (<http://www.resources.arcgis.com>).

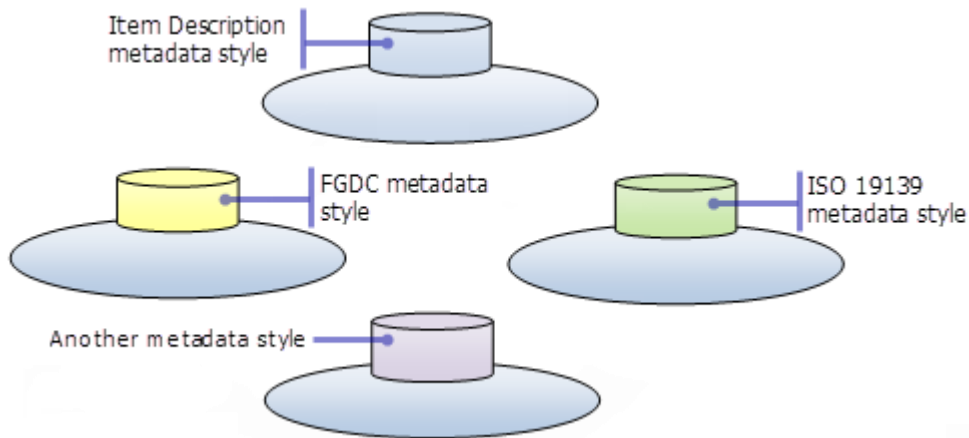


**Figura 7.** Contenido de los metadatos en ArcGis

No importa el estilo que se use toda la información proporcionada en la pestaña Descripción es almacenada en cada uno de los metadatos del elemento. De manera predeterminada cuando se visualiza los metadatos de un elemento ArcGis registra automáticamente la información de las propiedades intrínsecas de los metadatos en un proceso conocido como sincronización, esta propiedades solo son almacenadas y actualizadas en el formato de metadatos de ArcGis (<http://www.resources.arcgis.com>).

Elegir un estilo de metadatos es como aplicar un filtro a los metadatos de ArcGIS. Para los estilos de metadatos diseñados para admitir un estándar o un perfil, el filtro es el encargado de garantizar que se está trabajando con el contenido apropiado y con la información importante para ArcGis. Los estilos de metadatos configuran ArcGis para generar los metadatos que desee, el estilo controla la forma en que se pueden visualizar los metadatos y las páginas que aparecen para la edición de metadatos. Un estilo de metadatos puede estar diseñado para admitir un estándar o perfil de metadatos; si es así el estilo determina cuantos metadatos se exportan y validan para ese estándar o perfil. El estilo de metadatos descripción del elemento, es directo y efectivo, ideal para aquellos que no se adhieran a estándares de metadatos específicos (<http://www.resources.arcgis.com>).





**Figura 8.** Estilos de los metadatos en ArcGis

ArcGis ofrece diferentes estilos de metadatos, permitiendo administrar a sus metadatos correctamente para distintos estándares o perfiles. Dentro de los estilos de ArcGis encontramos las hojas de estilo XSLT, estas se pueden utilizar para realizar una gran variedad de modificaciones a metadatos ArcGis o a un archivo XML. Varias hojas de estilo XSLT se proporcionan con ArcGis Desktop. Se pueden encontrar en la carpeta Ubicación de instalación de ArcGIS\Metadatos\Stylesheets. En la siguiente tabla se describen estas hojas de estilo ([http:// www.resources.arcgis.com](http://www.resources.arcgis.com)).

**Tabla 3.** Hojas de estilo XSLT en ArcGis (<http://www.resources.arcgis.com>).

ArcGIS.xsl	Muestra el contenido completo de los metadatos de un elemento.
ArcGIS_ItemDescription. Xsl	Solo muestra una descripción concisa del elemento.

Las siguientes hojas de estilo que se encuentran en la carpeta ArcGIS Installation Location\Metadatos\Stylesheets\gpTools son las encargadas de tomar los metadatos de un elemento, procesarlas y finalmente guardan el documento XML resultante en un nuevo archivo XML. El objetivo es filtrar los metadatos de un elemento antes de ser utilizados fuera de ArcGis, o en su defecto modificar los metadatos del elemento en el caso de que el archivo XML resultante se pueda guardar nuevamente en el elemento original como sus

metadatos utilizando la herramienta importador de metadatos. En la siguiente tabla se mostraran las hojas de estilo que se proporcionan con ArcGIS for Desktop para realizar algunas tareas de metadatos bien definidas ([http:// www.resources.arcgis.com](http://www.resources.arcgis.com)).

**Tabla 4.** Hojas de estilo en ArcGis for Desktop (<http://www.resources.arcgis.com>).

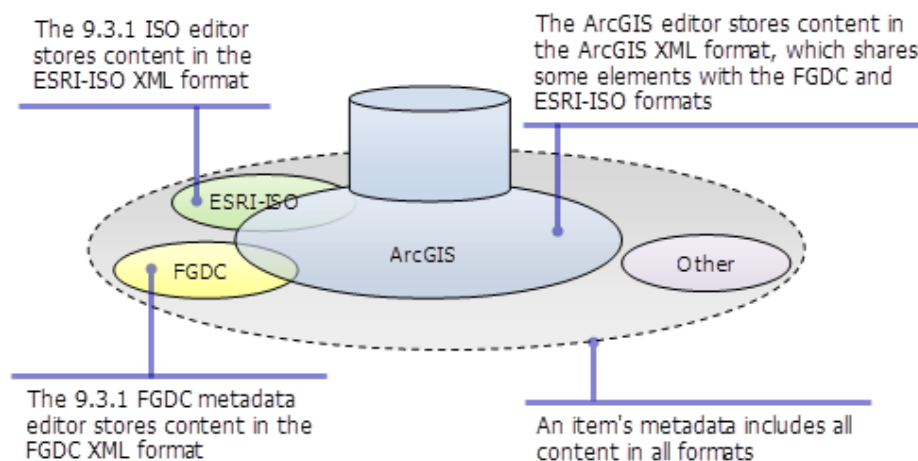
archivo agregar identificador único .xslt	Agrega o cambia el identificador único almacenado en los metadatos del elemento.
exact copy of.xslt	Crea una copia exacta de los metadatos del elemento como un archivo XML.
generate metadata template.xslt	Copia el contenido de los metadatos de ArcGIS a un nuevo archivo XML que se puede importar a otros elementos como plantilla de metadatos.
remove empty elements.xslt	Elimina los elementos XML vacíos.
remove entries from FGDC lineage.xslt	Elimina los pasos de proceso agregados al linaje de formato FGDC por ArcGIS 10.1 for Desktop 9.3.1.
remove FGDC required hints.xslt	Quita cualquier elemento de metadatos de formato FGDC que contenga el texto NECESARIO.
remove geoprocessing history.xslt	Quita el historial de geoprocesamiento de un elemento.
remove local storage info.xslt	Quita los nombres de equipo que pueden existir en los metadatos del elemento.
remove pre94 metadata elements.xslt	Quita los elementos XML del formato ESRI-ISO y cualquier elemento de formato XML.
remove synchronized elements.xslt	Quita cualquier información que se agregó a los metadatos de un elemento por el proceso de sincronización de metadatos de ArcGIS.
remove thumbnail.xslt	Quita la vista en miniatura de un elemento.
remove unique identifiers.xslt	Quita todos los identificadores únicos de los metadatos de un elemento.
_MPXML2.xsl	Extrae los elementos XML del formato FGDC <i>CSDGM</i> de los metadatos del elemento, si es que existen, y los ordena correctamente.

Por otra lado a parte de los estilos encontramos los elementos XML de metadatos de ArcGis, si se visualizan o editan en la pestaña Descripción esos metadatos solo contendrán

información almacenada en el formato de metadatos ArcGis. Sin embargo si encontramos un elemento que existió durante un tiempo o sus metadatos fueron creados utilizando la versión ArcGIS 9.3.1 for Desktop o una versión anterior. Probablemente el contenido de sus metadatos serán almacenados en los formatos XML de metadatos de FGDC o ESRI-ISO junto a otra información interna de ArcGIS ([http:// www.resources.arcgis.com](http://www.resources.arcgis.com)).

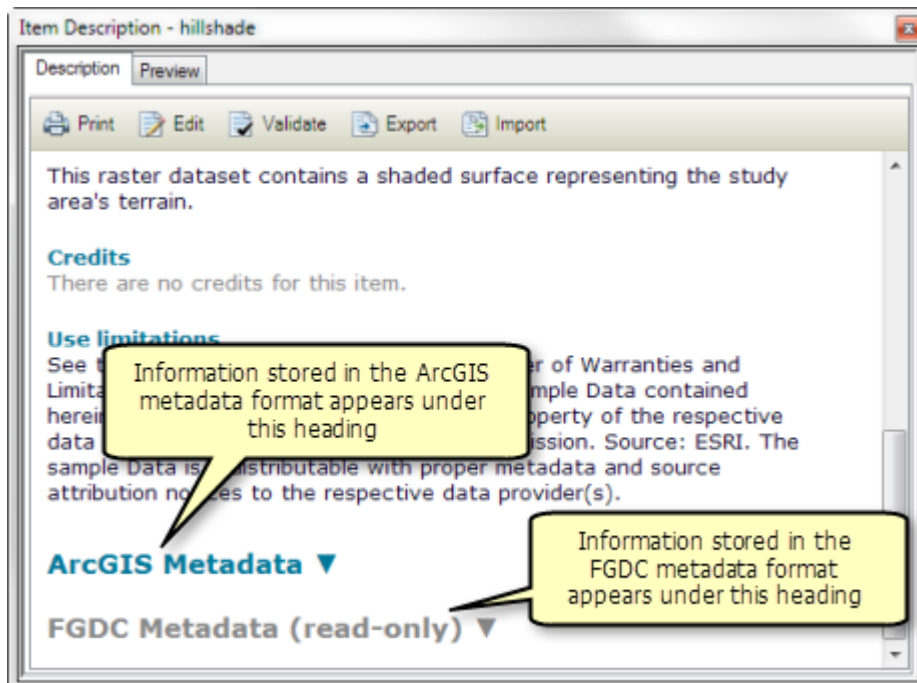
Tanto el editor como las utilidades de metadatos 9.3.1 FGDC crearon, mostraron y administraron el contenido en el formato XML de metadatos de FGDC nativo dentro de los metadatos encontrados en un elemento. Mientras que el editor y las utilidades de metadatos ISO estilo asistente 9.3.1 realizaron lo mismo que el anterior pero en formato XML de ESRI-ISO. Los elementos XML del formato FGDC y los elementos XML del formato ESRI-ISO son únicos de cada formato. El admitir otros estándares ha implicado agregar nuevas utilidades y editores personalizados para crear, visualizar y administrar el contenido, esto generalmente administrará el contenido en conjuntos adicionales de elementos XML paralelos ([http:// www.resources.arcgis.com](http://www.resources.arcgis.com)).

El formato de metadatos en ArcGis presta elementos XML del formato XML de FGDC, elementos que describen los atributos de un elemento. También toma prestado elementos XML incluidos en el formato XML de ESRI-ISO. Sin embargo los metadatos de ArcGis incluyen nuevos elementos XML también y a su vez almacena información de una nueva manera.



**Figura 9.** Almacenamiento de los metadatos en ArcGis

Cuando se visualizan los metadatos, los nuevos elementos de metadatos de ArcGis sincronizados se agregan automáticamente a cualquier contenido de metadatos con formato de FGDC o ESRI-ISO. También se puede observar en el formato de FGDC en el encabezado Metadatos de FGDC (solo lectura); este apartado o encabezado aparece si existen metadatos en el formato FGDC. Tanto el contenido entre el formato FGDC y el formato de ArcGis aparecerá en ambos encabezados ([http:// www.resources.arcgis.com](http://www.resources.arcgis.com)).



**Figura 10.** Visualización de los metadatos en ArcGis

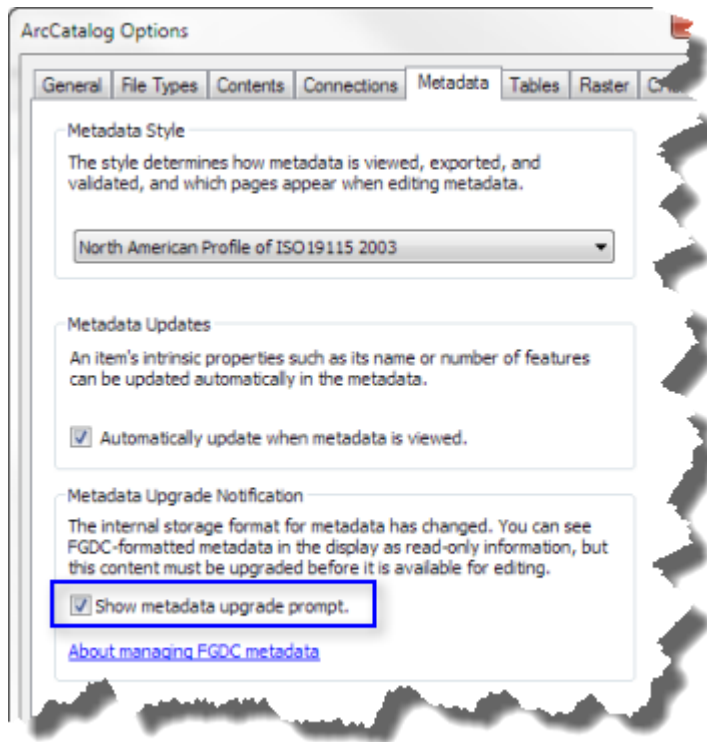
Se puede visualizar el contenido que se encuentra en formato FGDC, esta información esta solo disponible en la pestaña Descripción, la cual es la razón por la que la información se identifica solo como lectura.

Dentro de ArcGis encontramos la opción de actualizar los metadatos, esto hace referencia si un elemento tiene contenido de metadatos existente que no está en el formato de metadatos de ArcGIS, la versión actual de ArcGIS for Desktop no puede utilizar esa información. No hay necesidad de volver a escribir todos los metadatos existentes de un elemento en la pestaña Descripción. Existe una herramienta de geoprocésamiento que se encuentra

disponible para actualizar el contenido de metadatos con formato FGDC o el contenido de metadatos con formato ESRI-ISO ya existentes al formato de metadatos de ArcGIS, es la llamada **Actualizar metadatos**, copia información en los elementos de metadatos FGDC o ESRI-ISO existentes que no están incluidos en el formato de metadatos de ArcGIS a los elementos de metadatos de ArcGIS equivalentes ([http:// www.resources.arcgis.com](http://www.resources.arcgis.com)).

Cuando utilizamos la herramienta Actualizar, esta no altera el contenido interno de ArcGIS del elemento: el historial de geoprocésamiento, vistas en miniatura, adjuntos y así sucesivamente. Por otra parte tampoco elimina ningún elemento con formato FGDC- y ESRI-ISO existente. Las propiedades del elemento que fueron registradas en sus metadatos a través de ArcGIS for Desktop 9.3.1 no se actualizan. La versión actual de ArcGIS actualiza de forma automática los metadatos del elemento para incluir sus propiedades actuales al final del proceso de actualización ([http:// www.resources.arcgis.com](http://www.resources.arcgis.com)).

Si se desea actualizar el contenido de metadatos FGDC de inmediato. Existe la herramienta Actualizar metadatos se abre y Actualizar tipo se establece automáticamente en FGDC\_TO\_ARCGIS. Mientras por el contrario se prefiere esperar, se debe hacer click en no. El botón Actualizar estará disponible en la pestaña Descripción. Puede utilizarlos después para actualizar los metadatos del elemento. En ArcGis encontramos la opción para que no se avise cuando el contenido de metadatos de FGDC no se ha actualizado, seleccionando la opción No mostrar este mensaje en el futuro ([http:// www.resources.arcgis.com](http://www.resources.arcgis.com)).



**Figura 11.** Actualización de los metadatos en ArcGis

Finalmente después de crear, editar, validar y exportar metadatos a unos cuantos elementos, se empieza a buscar formas de sistematizar su flujo de trabajo. Se puede crear documentos de metadatos de plantilla que contengan información común para varios elementos. Aparte de la edición todas las tareas de metadatos se pueden lograr usando las herramientas de geoprocésamiento disponibles en la caja de herramientas de conversión, se puede importar una plantilla de metadatos o crear una secuencia de comandos Python apropiada. Por ejemplo, puede actualizar metadatos para todos los elementos en una carpeta o geodatabase. Como fue el caso de este proyecto ([http:// www.resources.arcgis.com](http://www.resources.arcgis.com)).

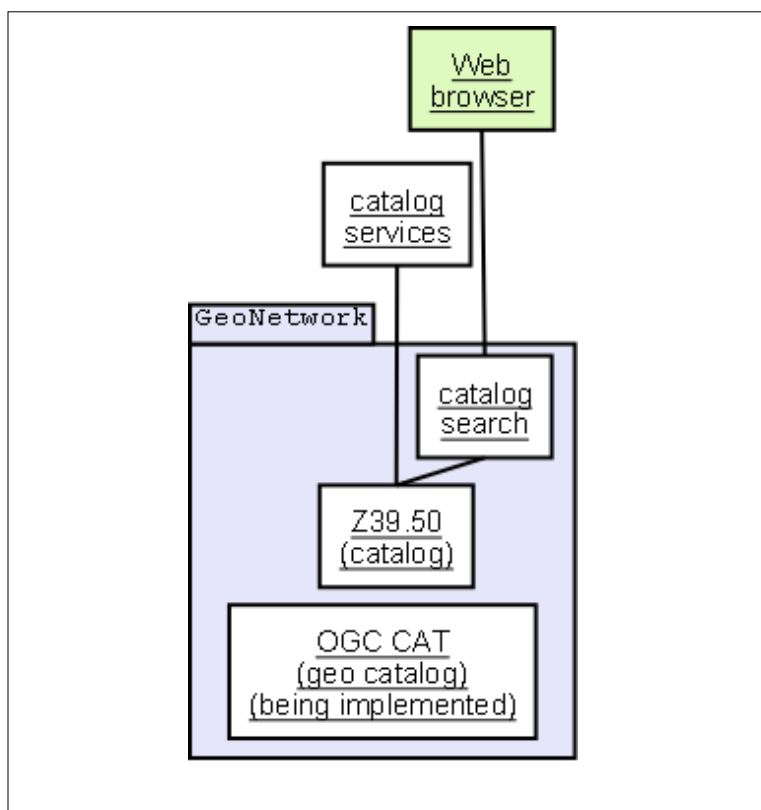
### 3.2 Geonetwork

Geonetwork, es una aplicación informática de software libre y código abierto de catalogación para recursos al espacio geográfico. Este es un entorno de información espacial estandarizado y descentralizado, el cual permite el acceso a bases de datos, productos cartográficos y metadatos originados de varias fuentes, con el fin de intercambiar y compartir entre las organizaciones y los usuarios a través del internet. En Geonetwork encontramos el protocolo Z39.50, usando este protocolo podemos acceder a

catálogos remotos y permitir que los datos estén disponibles para otros servicios de catálogos. En 2007 se implementaron OGC y Web Catalog Service (<http://geonetwork-opensource.org/>).

Este software ha sido implementado en varias organizaciones. Las primeras fueron la Geonetwork de la FAO y WFP VAM-SIE-Geonetwork, en Roma, Italia. También a OMS, CGIAR, BRGM, AEE, FGDC y el Global Change Information and Research Centre (GCIRC) de China quienes están trabajando en la implementación de Geonetwork opensource como su herramienta de gestión de información espacial (<http://geonetwork-opensource.org/>).

Existen varias herramientas que vienen en el paquete de Geonetwork, entre ellas Geoserver la cual almacena datos geográficos, mientras que Geonetwork cataloga colecciones de dichos datos.



**Figura 12.** Interfaces de Geonetwork (Enciclopedia Wikipedia.org)

Geonetwork Opensource también es definido en su portal como una aplicación de catálogo que gestiona los recursos espacialmente referenciados. Proporcionan por otra parte edición y búsqueda de metadatos de gran alcance y actualmente es utilizado en numerosas infraestructuras de datos espaciales de todo el mundo (<http://geonetwork-opensource.org/>).

Geonetwork ha sido desarrollada para intercambiar información espacial y sus datos bajo una arquitectura moderna, la cual es potente y de bajo costo , se basa en software libre y de código abierto (FOSS) como ya habíamos mencionado anteriormente, y estándares internacionales y abiertos para servicios y protocolos de ISO, TC211 Y OGC (**GeoNetwork opensource**, 2007-2009).

El software proporciona una manera fácil y eficaz de utilizar la interfaz web para la búsqueda de datos geoespaciales a través de múltiples catálogos. También se encarga de combinar los servicios de mapas, publicar datos geoespaciales, esto utilizando los metadatos encontrados en línea. Por otro lado los administradores tienen la posibilidad de administrar cuentas de usuarios y de grupo (<http://geonetwork-opensource.org/>).

Algunas de las características principales son:

- Acceso inmediato local y distribución geoespacial de los catálogos.
- Carga y descarga datos, documentos, archivos pdf y cualquier otro tipo de contenido.
- Visor de mapa interactivo, para combinar servicios web de mapas distribuidos en todo el mundo.
- Permite la edición de metadatos en línea, mediante un potente sistema de plantillas que posee.
- Recolección programada y sincronización de metadatos entre catálogos distribuidos.
- Control de acceso con el grupo y la gestión de usuarios.
- Multidioma de interfaz de usuario.



La comunidad de Geonetwork se ha ido expandiendo durante los últimos años; esto ha sido posible a todos los usuarios que han contribuido a través de código, pruebas, informes de errores, correcciones y demás sugerencias.

En el 2001 La Organización para la Alimentación y Agricultura de Naciones Unidas (FAO) desarrolló el prototipo de GeoNetwork para archivar y publicar sistemáticamente los juegos de datos geográficos producidos en dicha organización. Utilizaba metadatos de sistemas antiguos, transformados en el único borrador de metadatos estándar disponible en aquel entonces, el ISO 19115 (**GeoNetwork opensource**, 2007-2009).

En 2004, La siguiente versión permitió a los usuarios trabajar con varios estándares de metadatos (ISO 19115, FGDC y Dubli Core) de forma transparente. Por otro lado permitió compartir metadatos entre varios catálogos mediante un mecanismo de cacheo, mejorando la fiabilidad en las búsquedas a través de múltiples catálogos (**GeoNetwork opensource**, 2007-2009).

En el 2007, con una nueva versión. El soporte para el estándar ISO19115:2003 ha sido posible utilizando la implementación del esquema de la especificación ISO19139:2007 publicado en mayo. Esta versión sirve como implementación libre de referencia de la especificación del Servicio de Catálogo para Web (CSW 2.0.1) de OGC (**GeoNetwork opensource**, 2007-2009).

Cada año se ha ido mejorando el desarrollo de Geonetwork dando a los usuarios nuevas experiencias de uso más interactivo, incluyendo un nuevo visor de mapas y una revisión completa de la interfaz de búsqueda.

### **3.2.1 Búsquedas en Geonetwork**

Geonetwork cuenta con tres diferentes métodos de búsqueda: por defecto, avanzado y por categoría, cada uno de ellos proporcionan una lista de cada uno de los registros de metadatos los cuales cumplen los criterios de su búsqueda. Para cada registro encontrado

los resultados muestran el título, el resumen y las palabras clave. Se pueden consultar un máximo de cuatro secciones, en función de los privilegios que se han establecido para cada metadato (**GeoNetwork opensource, 2007-2009**).

Entre las clases de búsqueda encontramos la búsqueda por defecto permite la búsqueda de texto en cualquier lugar del registro, como palabras clave del metadato y/o la localización geográfica. El texto y los operadores no son sensibles a mayúsculas ni a minúsculas (**GeoNetwork opensource, 2007-2009**).



**Figura 13.** Búsqueda por defecto en Geonetwork

La segunda clase de búsqueda avanzada, esta es más específica en sus criterios de búsqueda ya que se ofrecen más elementos por los que buscar información, cada uno de ellos enfocado a uno de los aspectos siguientes: ¿Qué?, ¿Dónde?, ¿Cuándo? (**GeoNetwork opensource, 2007-2009**).



**Figura 14.** Búsqueda avanzada en Geonetwork

Finalmente búsqueda por categoría, simplemente hay que destacar la categoría por la que quiere realizar la búsqueda en la lista, en otro caso, se puede dejar este campo en Cualquier Categoría (GeoNetwork opensource, 2007-2009).

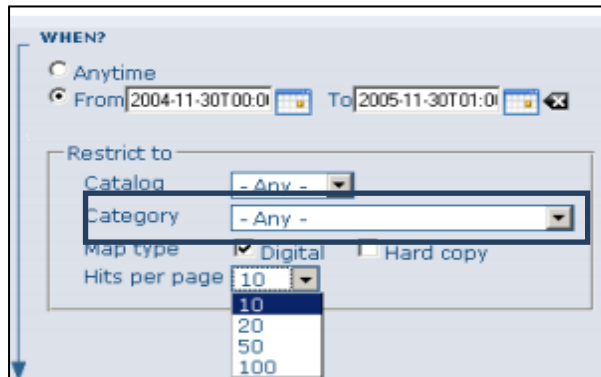


Figura 15. Búsqueda por categoría en Geonetwork

Por otra parte al realizar la búsqueda en la sección de metadatos, donde se describe el juego de datos y puede contener enlaces a otros sitios web que proporcionar más información sobre el juego de datos.



Figura 16. Búsqueda en la sección de metadatos en el portal de Geonetwork

La descripción de metadatos; el perfil de metadatos ISO 19139 usado en GeoNetwork opensource describe datos geográficos y servicios se basa en el estándar internacional ISO 19115:2003, codificado de acuerdo al esquema de implementación 19139:2007, el estándar FGDC y el estándar internacional Dublin Core. Proporciona información relacionada con la identificación, mantenimiento y restricciones, la extensión temporal y espacial, la representación espacial y referencia.

La sintaxis, es parte del lenguaje que provee de medios para representar las estructuras, entrega mecanismos para codificar, intercambiar, visualizar y procesar los metadatos. Un ejemplo es XML. Otro elemento no funcional dentro del proyecto es el lenguaje que fue utilizado para la realización de este, como ya se ha mencionado en ítems anteriores, Python fue el lenguaje utilizado para la programación, este es un lenguaje interpretado o de script, en este caso el desarrollo del proyecto se hizo mediante un script.

```

74 Script_final_17Dic.py - D:\Master LIGIT\Proyecto Final LIGIT\Proyecto Final LIGIT\Versiones script final\Script_final_17Dic.py
File Edit Format Run Options Windows Help
#Import modules
import arcpy, sys, os
from arcpy import env
import xml.etree.ElementTree as ET
import traceback

BaseDataPathURL="file:///etc/sial2/Reference_Data"
BaseDataPathWin="\\\\etc/sial2\\Reference_Data"
#MetadataOutPath = "F:\\ISOMETADATA"

# --- setProjection.py ---
# Sets the projection for all the layers contained in a folder
# AND SUBFOLDERS (take care!!!).
# It works for both raster and vector layers.
# Params:
# inWorkspace: the base folder to apply the script
# [projectionCode] the code of the projection to apply. By default,
# code 3035 (ETRS89 LAEA) is applied
# -----

def processFolder(folder, recurseDirectories=True):
    logging.info(" * Processing folder: " + folder + "...")
    print " * Processing folder: " + folder + "..."
    folderbak = env.workspace
    env.workspace = folder
    doExportFC(folder)
    doExportDataset(folder)
    if (recurseDirectories):
        wpList = arcpy.ListWorkspaces("", "ALL")
        for wp in wpList:
            processFolder(wp, recurseDirectories)
    env.workspace = folderbak

def insertDataPath(datapath, isoxml):
    print "HOLA" + isoxml
    tree = ET.parse(isoxml)
    root = tree.getroot()

    ET._namespace_map["http://www.isotc211.org/2005/gmd"] = "gmd"
    ET._namespace_map["http://www.isotc211.org/2005/gco"] = "gco"
    ET._namespace_map["http://www.isotc211.org/2005/gs"] = "gs"

```

Figura 17. Script lenguaje Python

Finalmente son varias las compañías que ofrecen el soporte comercial para ayuda a las diferentes organizaciones a implementar, integrar y mantener el software.

### 3.3 Lenguaje Python

Es un lenguaje de programación creado por Guido van Rossum en los años 90, su nombre fue inspirado en el grupo cómicos ingleses “Monty Python”; este lenguaje presenta una sintaxis limpia y de código legible (**Duque González, R**).

Python es un lenguaje interpretado o de script, multiplataforma y orientado a objetos. El lenguaje es aquel que se ejecuta utilizando como su nombre lo indica un programa llamado intérprete. En Python como en muchos otros lenguajes, el código fuente se traduce a un pseudo código máquina intermedio denominado bytecode la primera vez que se ejecuta genera archivos .pyc o .pyo (**Duque González, R**).

El tipado con el que cuenta Python es dinámico, se refiere a que no es necesario el tipo de dato que va contener una determinada variable, si no que su tipo se determinara en tiempo de ejecución según el tipo de valor que se asigne y el tipo de la variable cambia si se le asigna un valor de otro tipo. En este lenguaje a diferencia de otros si tenemos una variable que contiene texto no podremos tratarla como un número, mientras que en otros lenguajes el tipo de variable cambiara para adaptarse al comportamiento esperado.

El intérprete de Python se encuentra disponible en multitud de plataformas; si no utilizamos librerías específicas de cada una de las plataformas, el programa correrá en todas los sistemas sin grandes cambios.

Es un lenguaje que va dirigido a la orientación de objetos, ya que la ejecución del programa consiste en una serie de interacciones entre los objetos. Python permite programación imperativa, funcional y orientada a aspectos.

Python es un lenguaje que todo el mundo debería manejar y conocer, que la sintaxis es clara, simple y sencilla; tipado dinámico, posee una gran cantidad de librerías disponibles, y hace que el desarrollo de una determinada aplicación sea sencillo y lo más importante de manera rápida. Algunos ejemplos donde python ha actuado con éxito son Google, Yahoo, La Nasa, Industrias Light y Magis, y todas las distribuciones Linux (**Downey, et al** 2002).

Existen varias implementaciones distintas de Python: CPython, Jython, IronPython, PyPy. CPython esta última, es la más utilizada, la más rápida y la más madura. Está instalado por defecto en la mayor parte de las distribuciones Linux y en las últimas versiones de Mac OS. Jython es la implementación en Java de Python, mientras que IronPython es su contrapartida en C# (.NET). Su interés estriba en que utilizando estas implementaciones se pueden utilizar todas las librerías disponibles para los programadores de Java y .NET. PyPy, por último, se trata de una implementación en Python de Python (**Downey, et al** 2002).

Para la ejecución de Python existen dos formas de ejecutar código, en una sesión interactiva (línea a línea) con el intérprete, o bien de la forma habitual, escribiendo el código en un archivo de código fuente y ejecutándolo. Para la ejecución de cualquier programa con el lenguaje Python debe indicársele el nombre del archivo a ejecutar al intérprete de Python. Ejemplo: Nombre del archivo.py (**Duque González, R**).

Si por el contrario utilizamos Windows los archivos.py ya se encuentran asociados al intérprete de Python así que solo basta con hacer doble click, este imprimirá un texto en la consola y su ejecución es demasiado rápida para visualizarla, para solucionar esto se añade una nueva línea donde se espera la entrada de datos por parte del usuario (**Duque González, R**).

Si se utiliza Linux, para que el sistema operativo abra el archivo .py con el intérprete adecuado, es necesario añadir una nueva línea al principio del archivo:

`#!/usr/bin/python` A esta línea se le conoce en el mundo Unix como *shebang*, *hashbang* o *sharpbang*. El par de caracteres que se presentan en la línea anterior ( `#!`) indican al sistema operativo que dicho script debe ejecutar.

En Python los tipos básicos se dividen en: Números, como pueden ser entero (3), flotante (15.57) o complejos ( $7 + 5j$ ); cadenas de texto, como “Hola Mundo” y valores booleanos conocidos como True (cierto) y False (falso) (**Duque González, R**).

En Python encontramos algunos tipos de colecciones de datos como listas, tuplas y diccionarios.

Las listas son un tipo de colecciones ordenadas. En otros lenguajes es lo que se conoce por arrays o vectores. Estas pueden contener cualquier tipo de dato sea números, cadenas, booleanos. Es de forma sencilla la creación de estas listas se realiza entre corchetes y se separa por comas cada uno de los valores que se quiere incluir en la lista (**Duque González, R**).

A las tuplas se aplica lo mismo de las listas, solo cambia la forma de definirla, para lo que se utilizan paréntesis en lugar de corchetes. En realidad el constructor de la tupla es la coma, no el paréntesis.

Finalmente los diccionarios también llamados matrices asociativas, deben su nombre a que son colecciones que relacionan una clave y un valor.

En el lenguaje Python encontramos el control de flujo, esto hace referencia a los condicionales y los bucles, como su nombre lo indica los condicionales nos permiten comprobar condiciones y hacer que nuestro programa se comporte de una forma u otra, que ejecute un fragmento de código u otro, dependiendo de esta condición. Es aquí donde cobran importancia el tipo booleano y los operadores lógicos. Los bucles nos permiten ejecutar un mismo fragmento de código, un cierto número de veces, mientras se cumpla una determinada condición; a diferencia de los condicionales que nos permiten ejecutar distintos fragmentos de código dependiendo de ciertas condiciones (**Duque González, R**).

Las funciones en el lenguaje Python es un fragmento de código con un nombre asociado que realiza una serie de tareas y devuelve un valor. A los fragmentos de código que tienen un nombre asociado y no devuelven valores se les suele llamar procedimientos. En Python no existen los procedimientos, ya que cuando el programador no especifica un valor de retorno la función devuelve el valor None (nada), equivalente al null de Java (**Duque González, R**).

Además de ayudar a programar y depurar dividiendo el programa en partes las funciones también permiten reutilizar código.

Como habíamos mencionado Python es un lenguaje multiparadigma en él se podía trabajar con programación estructurada, o con programación orientada a objetos o programación funcional.

La Programación Orientada a Objetos (POO u OOP según sus siglas en inglés) es un paradigma de programación en el que los conceptos del mundo real, relevantes para nuestro problema se modelan a través de clases y objetos. Para entender primero tenemos que comprender qué es una clase y qué es un objeto: Una clase, no es más que una plantilla genérica a partir de la cuál instancian los objetos; plantilla que es la que define qué atributos y métodos tendrán los objetos de esa clase (**Duque González, R**).

Un objeto es una entidad que agrupa un estado y una funcionalidad relacionada. El estado del objeto se define a través de variables llamadas atributos, mientras que la funcionalidad se modela a través de funciones a las que se les conoce con el nombre de métodos del objeto.

Para finalizar el concepto de lo que es el lenguaje Python, encontramos que existen tres conceptos que son básicos para cualquier lenguaje de programación orientado a objetos: el encapsulamiento, la herencia y el polimorfismo.

En un lenguaje orientado a objetos cuando hacemos que una clase (subclase) herede de otra clase (superclase) estamos haciendo que la subclase contenga todos los atributos y métodos que tenía la superclase. No obstante al acto de heredar de una clase también se le llama a



menudo “extender una clase”. En Python, a diferencia de otros lenguajes como Java o C#, se permite la herencia múltiple, es decir, una clase puede heredar de varias clases a la vez (**Duque González, R**).

El polimorfismo, se refiere a la habilidad de objetos de distintas clases de responder al mismo mensaje. Esto se puede conseguir a través de la herencia: un objeto de una clase derivada es al mismo tiempo un objeto de la clase padre, de forma que allí donde se requiere un objeto de la clase padre también se puede utilizar uno de la clase hija. En ocasiones también se utiliza el término polimorfismo para referirse a la sobrecarga de métodos, término que se define como la capacidad del lenguaje de determinar qué método ejecutar de entre varios métodos con igual nombre según el tipo o número de los parámetros que se le pasa. En Python no existe sobrecarga de métodos (el último método sobrescribiría la implementación de los anteriores), aunque se puede conseguir un comportamiento similar recurriendo a funciones con valores por defecto para los parámetros o a la sintaxis `*params` o `**params` (**Duque González, R**).

La encapsulación se refiere a impedir el acceso a determinados métodos y atributos de los objetos estableciendo así qué puede utilizarse desde fuera de la clase.

En Python no existen los modificadores de acceso, y lo que se suele hacer es que el acceso a una variable o función viene determinado por su nombre: si el nombre comienza con dos guiones bajos (y no termina también con dos guiones bajos) se trata de una variable o función privada, en caso contrario es pública. Los métodos cuyo nombre comienza y termina con dos guiones bajos son métodos especiales que Python llama automáticamente, bajo ciertas circunstancias (**Duque González, R**).

### **3.3.1 Python en ArcGis**

Python es un lenguaje gratuito, multiplataforma y de código abierto el cual es potente y fácil de aprender. Es ampliamente utilizado y compatible.

Python fue presentado a la comunidad de ArcGis en la versión 9.0. Desde entonces fue aceptada como el lenguaje de secuencia de comandos de elección para el geoprocésamiento de usuarios. Python se extiende a través de ArcGis y se ha convertido en el lenguaje para

análisis, conversión, administración de datos y automatización de mapas, lo que ha ayudado a aumentar la productividad (<http://www.help.arcgis.com>).

Esri ha aceptado completamente a Python para ArcGis, ya que lo ve como un lenguaje que satisface las necesidades de los diferentes usuarios, y muestra algunas ventajas de Python en ArcGis (<http://www.help.arcgis.com>).

- Es fácil aprender y excelente para personas que inician por primera vez con el manejo de este lenguaje, pero también es adecuado para personas expertas.
- Altamente escalable, adecuado para grandes proyectos o para pequeños programas de uso único conocidos como secuencias de comando.
- Portátil, multiplataforma.
- Incorporable (Permite que ArcGis se pueda escribir en secuencias de comandos).
- Estable y maduro
- Presenta una gran comunidad usuarios que hacen uso de este lenguaje de programación.

### **3.3.2 ArcPy en ArcGis**

ArcPy es un paquete de sitio el cual se basa en el módulo `arcgisscripting` y lo sucede. Su objetivo es crear una manera útil y productiva de realizar análisis de datos geográficos, conversión, administración de datos y automatización de mapas con Python (<http://www.help.arcgis.com>).

Este paquete ofrece finalización de código, así como documentación de referencia para cada función, módulo y clase.

La ventaja de utilizar ArcPy dentro de Python es que Python es un lenguaje de programación del propósito general. Es un lenguaje interpretado con asignación dinámica de tipos, adecuada para el trabajo interactivo y la creación rápida de prototipos en programas únicos conocidos como secuencias de comandos, además ofrece suficiente potencia para permitir la escritura de aplicaciones grandes. Las aplicaciones ArcGis que se

escriben con ArcPy se benefician de modulos adicionales en numerosos nichos de Python por parte de profesionales del SIG y programadores de diferentes disciplinas (<http://www.help.arcgis.com>).

Python ofrece el servicio de cadenas de caracteres de documentación. Las funciones y clases disponibles en ArcPy utilizan este método para la documentación del paquete.

Para obtener ayuda es utilizar el comando help de Python, otro método para la ayuda es la finalización de código que proporciona ArcPy. Siempre que se escriba un nombre de función y comience a escribir argumentos, se mostrara la ayuda y el uso de la función en la ventana de ayuda (<http://www.help.arcgis.com>).

#### **4. ANÁLISIS DE REQUERIMIENTOS**

En este ítem debemos tener en cuenta lo que quiere el cliente, para ello se debe diagnosticar la situación actual, recopilando cada uno de los requerimientos del cliente, tanto en relación al sistema como generales es decir la situación ideal para poder definir alternativas de solución, según las cuales se puede avanzar desde lo que hoy se posee, hacia el punto que se pretende llegar.

##### **4.1 Requerimientos Funcionales**

1. Permitir la transformación de metadatos de formato ESRI XML, a formato estándar XML 19139, para la adición y publicación en el catálogo de Geonetwork.
2. Permitir la búsqueda y transformación de metadatos propagándose a todo el contenido de una carpeta raíz.
3. Permitir la búsqueda y la transformación de metadatos para todos los tipos de datos.
4. Los metadatos publicados en el catálogo de Geonetwork deben tener la ruta a los datos.

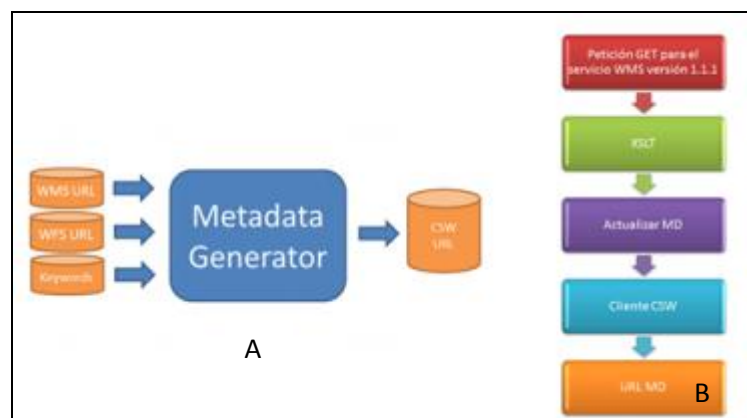
5. Permitir la búsqueda directa a los datos de los metadatos directamente desde el catálogo de Geonetwork.

#### 4.2 Requerimientos No Funcionales

1. El lenguaje de programación que se utilizó para realizar importación fue mediante Python.
2. Realización de script en Python con el entorno ArcGis, para transformaciones de formato, e incorporación a los metadatos la ruta directa a los datos desde el catálogo de Geonetwork.
3. Para acceder directamente a los datos desde el catálogo de Geonetwork, se realizó una configuración tanto de Internet Explorer como de Mozilla Firefox.

#### 4.3 Requerimientos de los Metadatos

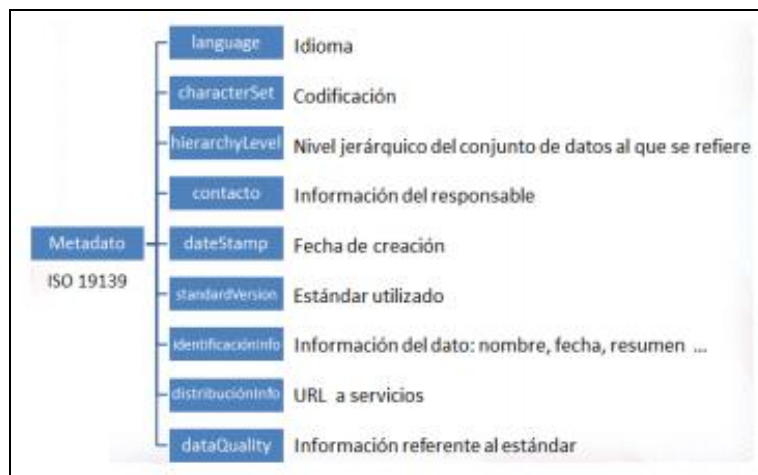
Proceso de generación de los metadatos, donde se observa las entradas y salidas del módulo *Metadata Generator* (Figura 18) (Oliver S, *et al* 2001).



**Figura 18.** A Parámetros de entrada y salida. B Secuencia de pasos de generador de metadatos

1. Lo primero que se realiza es una petición GetCapabilities a los servidores de datos donde ha sido previamente publicado.
2. El siguiente paso es aplicar la plantilla XSLT de generación de metadatos que se ha implementado. La transformación tiene como entrada el XML de la petición anterior. El resultado obtenido tras realizar la transformación es otro documento XML que forma el metadato cumpliendo con los estándares ISO 191139.
3. Por ultimo este paso consiste en parsear el XML del metadato, para rellenar los campos que no han sido posibles llenar durante la transformación.

Cabe mencionar que los metadatos generados no contienen todos los campos llenos de la especificación ya que algunos no se pueden obtener de forma automática. A continuación en la **Figura 19** mostraremos algunos de los campos que contiene el metadato generado. Por otro lado en una Tabla colocaremos los campos obligatorios que debe contener el metadato para ser publicado y validado en Geonetwork.



**Figura 19.** Algunos campos del metadato

Información necesaria para la publicación y validación de los metadatos en Geonetwork:

**Tabla 5.** Información básica para publicación y validación de metadatos en Geonetwork

<b>Items</b>	<b>Campos a diligenciar</b>
Item description	Titulo Resumen (summary) Descripción (Abstract) Créditos Uso de limitación
Citation	Titulo
Contacts	Contacto
Details	Crédito
Poitsn de contact	Contactos

## **5. METODOLOGÍA**

La metodología pretende hacer referencia a los procedimientos utilizados para alcanzar cada uno de los objetivos propuestos en el proyecto final. En este punto se mostrara el plan de trabajo que se realizó durante los tres meses de desarrollo de la memoria. El tratamiento de los datos y finalmente la recolección de la información, permitiendo el diseño final del proyecto.

### **5.1 Plan de Trabajo**

El proyecto final del master se realizó en un periodo de tres meses, durante los cuales se cumplieron 8 tareas.

La primera consta de la documentación de los diferentes conceptos y los programas que van a ser de uso para el proyecto. Entre los conceptos a tener en cuenta, se debe saber que es un metadato y de que consta, que son los archivos XML ISO19139, las herramientas como Python y Arcpy, y finalmente familiarizarse con Geonetwork.














En la segunda, se realizó un análisis del sistema que se iba a implementar

De la tercera a la sexta, se realizaron los diferentes script y el acoplamiento con Geonetwork, para esto el desarrollo fue el siguiente:

- Primero se desarrolla un script para exportar un metadato individual a formato XML 19139
- Segundo se realiza un estudio de la visualización de metadatos en Geonetwork para determinar el campo en el que se debería insertar la ruta a los datos.
- Tercero se desarrolla un segundo script que durante la exportación añade la ruta a los datos en el campo elegido en la tarea anterior.
- Finalmente se realiza un último script que exporte todos los metadatos de una estructura de directorios.

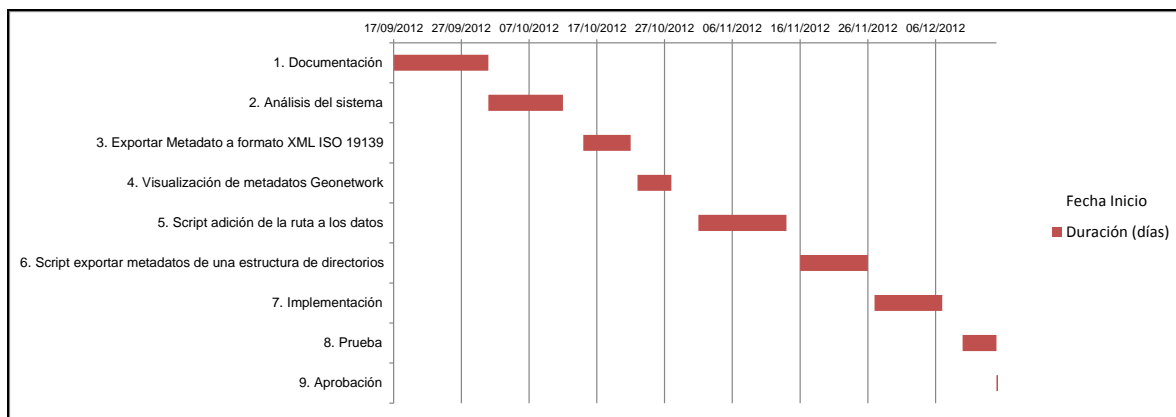
La séptima y octava fue la implementación y la prueba de la herramienta, con cada uno de los resultados obtenidos.

**Tabla 6.** Programación semanal para el desarrollo de las fases.

<b>SEPTIEMBRE 2012</b>	1Semana 17-21 	2Semana 24-28 					
<b>OCTUBRE 2012</b>	1Semana 1-5 	2Semana 8-12 	3Semana 15-19 	4Semana 22-26 	5Semana 29-31 		
<b>NOVIEMBRE 2012</b>	1Semana 5-9 	2Semana 12-16 	3Semana 19-23 	4Semana 26-30 			
<b>DICIEMBRE 2012</b>	1Semana 3-7 	2Semana 10-14 					



**Tabla 7.** Diagrama de Gannt



## 5.2 Tratamiento de los datos

El presente proyecto partió del análisis y de la revisión del catálogo de información geográfica del ETC-SIA/UAB.

Para cada objetivo específico se describió su método correspondiente y el análisis de cada uno de ellos.

Para este proyecto se utilizó como base el catálogo de información geográfica del Centro, donde fue encontrado cada uno de los metadatos registrados. Para el desarrollo de esta herramienta se llevaron a cabo las siguientes fases.



- Recolección de la información
- Instalación, personalización y configuración de un catálogo de metadatos (Geonetwork)
- Herramienta para la búsqueda de metadatos en formato ESRI XML y transformación a formato XML ISO 19139.
- Configuración de Geonetwork para añadir la ruta a los datos.

### 5.3 Recolección de la información

Se revisó los metadatos encontrados en el Centro, se verificó que los campos obligatorios estuvieran completamente diligenciados y en aquellos que no era el caso se procedió a rellenar los campos, posteriormente se realizó la transformación a formato XML ISO 19139, aunque el formato XML ISO 19139 no puede ofrecer un mecanismo de control absoluto sobre la validez de los metadatos, es el formato en cual los metadatos van a poder ser publicados y aceptado en Geonetwork.

#### Formato Esri XML:

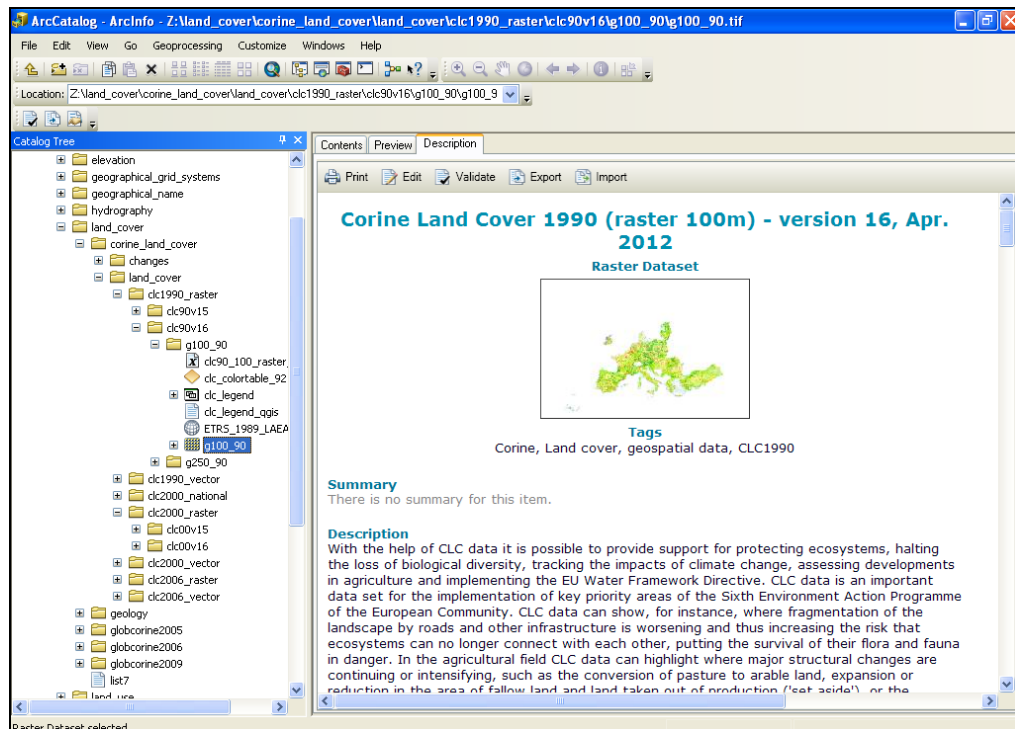
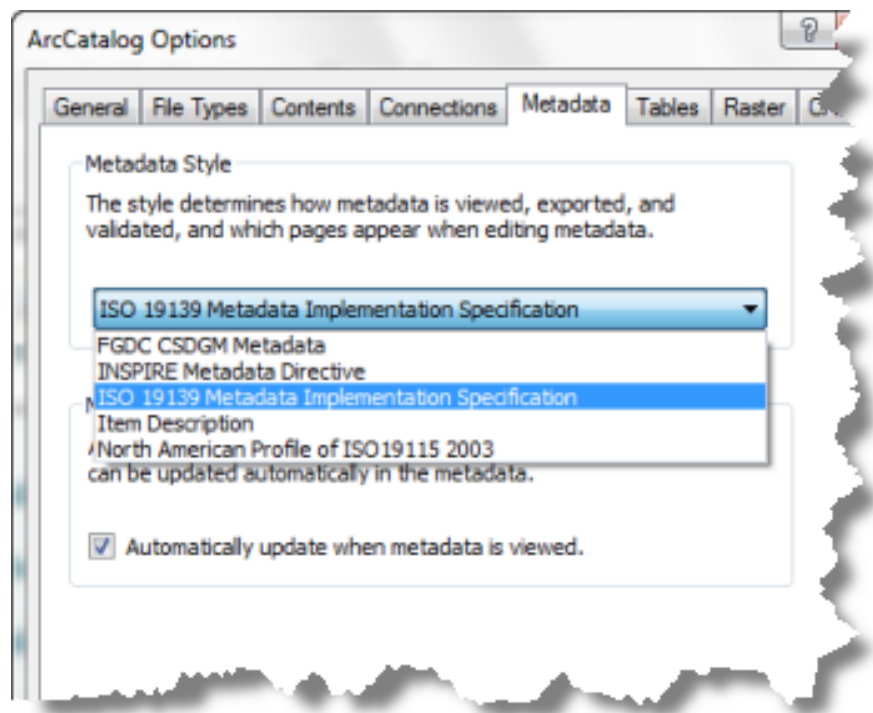


Figura 21. Descripción metadatos en ArcCatalog

Los estilos de metadatos configuran ArcGis para la generación de metadatos que se desee. Un estilo de metadatos esta diseñado para admitir un estandar o perfil de metadatos. Asi el estilo determinara cuantos metadatos se exportan y validan para ese estandar o perfil.

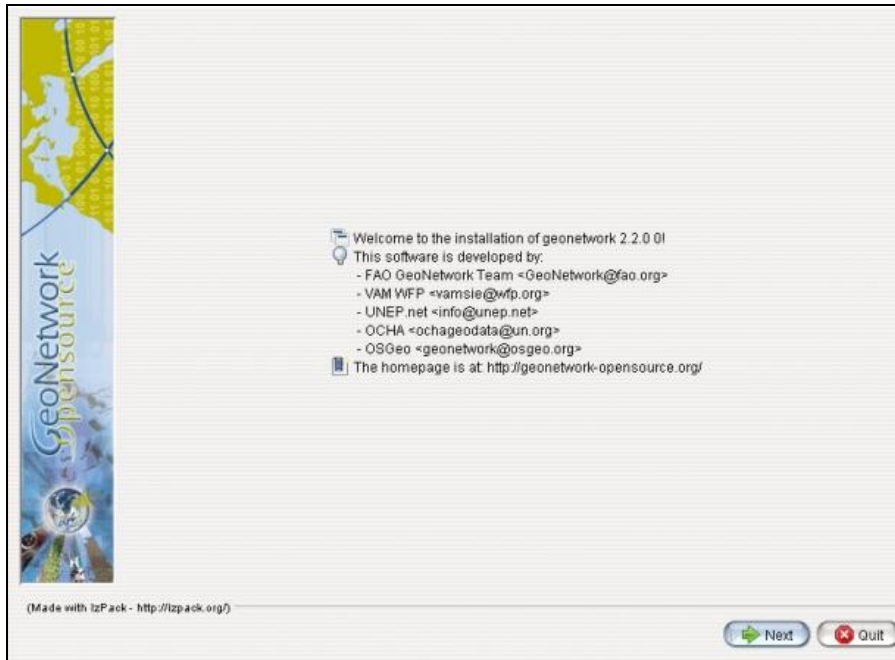
ArcGis ofrece diferentes clases de estilos, como se ve en la **Figura 22** donde se puede seleccionar el estilo que usted desee, permitiendole visulzar los elemento completos de un elemento.



**Figura 22.** Diversos estilos en ArcCatalog

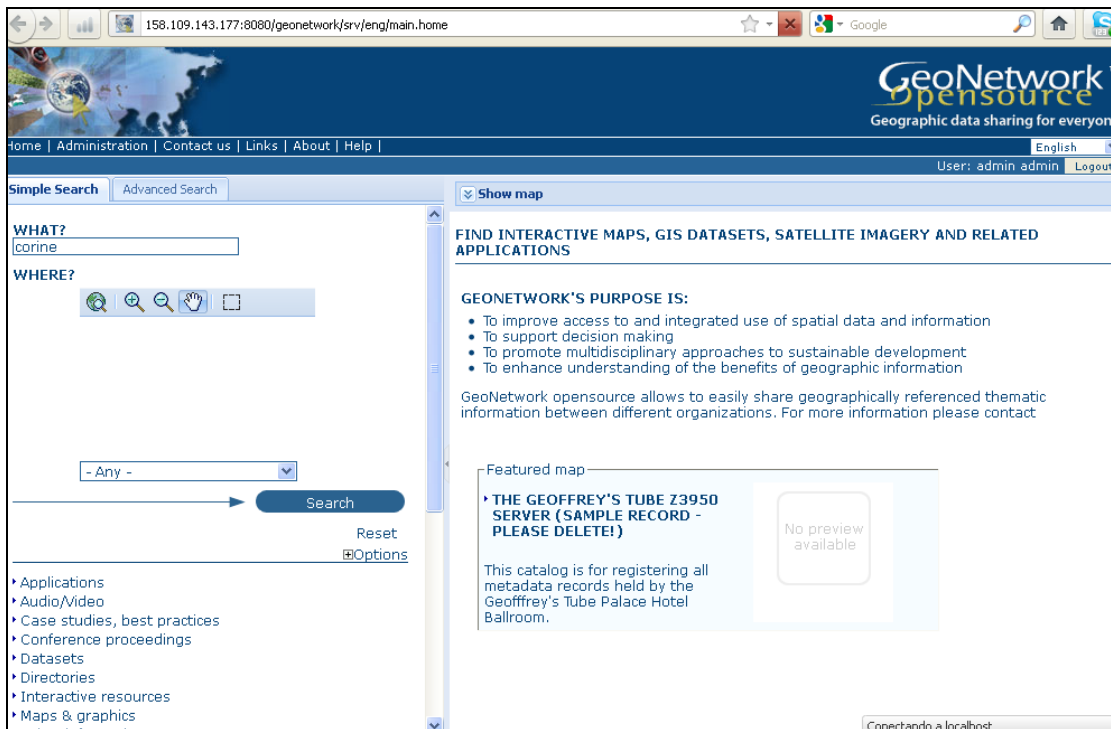
### **5.3.1 Instalación, personalización y configuración de un catálogo de Metadatos (Geonetwork)**

Para instalar Geonetwork se hace directamente de la página web. **Figura 23**



**Figura 23.** Instalación de Geonetwork

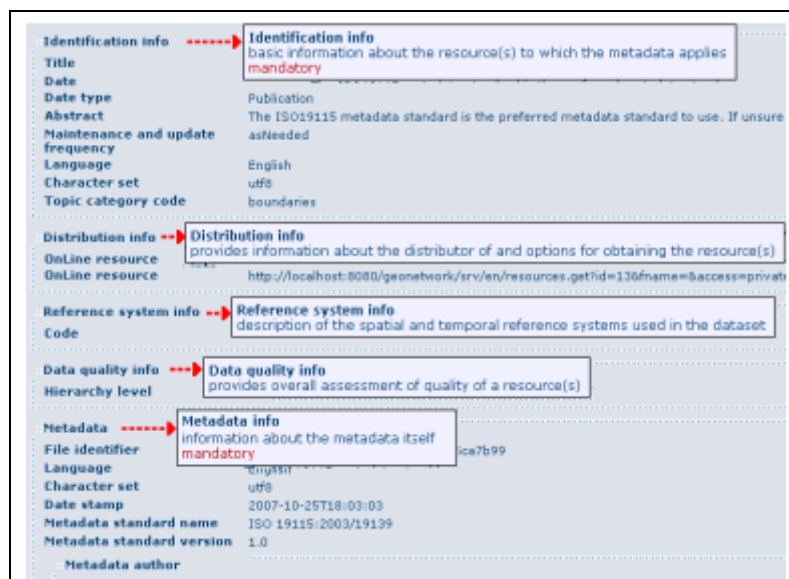
El catálogo de metadatos más importante es GeoNetwork **Figura 24.**



**Figura 24.** Portal del catálogo de Geonetwork

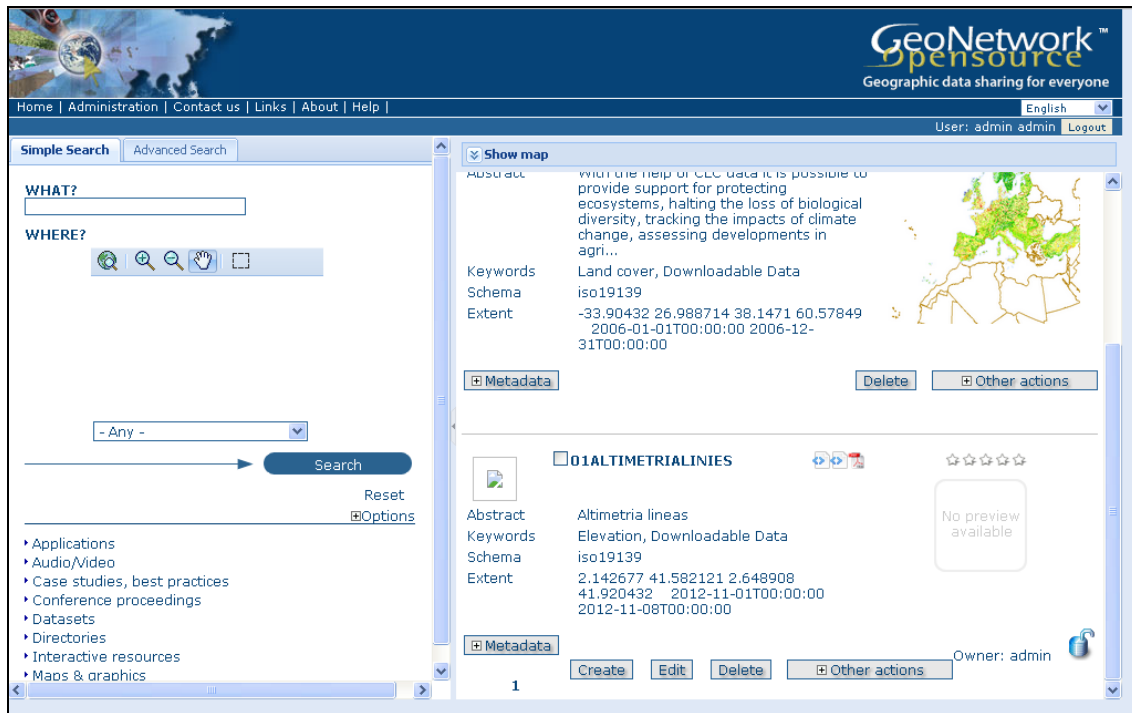
Los perfiles utilizados por Geonetwork para presentar y describir datos geográficos se basa en el Estándar Internacional ISO 19115:2003, codificado de acuerdo al esquema de implementación 19139:2007, el estándar FGDC y el estándar internacional Dublin Core (**GeoNetwork opensource**, 2007-2009).

El perfil de metadatos ISO 19139 usado en Geonetwork describe datos geográficos y servicios se basa en el estándar 19115:2003, proporciona información relacionada con la identificación, mantenimiento, restricciones, extensión espacial y distribución de un juego de datos geográficos (**GeoNetwork opensource**, 2007-2009). Organización de las secciones de metadatos en Geonetwork **Figura 25**.



**Figura 25.** Sección principal de metadatos en Geonetwork

Los metadatos basados en estándares aseguran la consistencia de los metadatos y su calidad, evitando que importantes partes del conocimiento de los datos se pierdan. A continuación se observa los metadatos publicados dentro de Geonetwork.

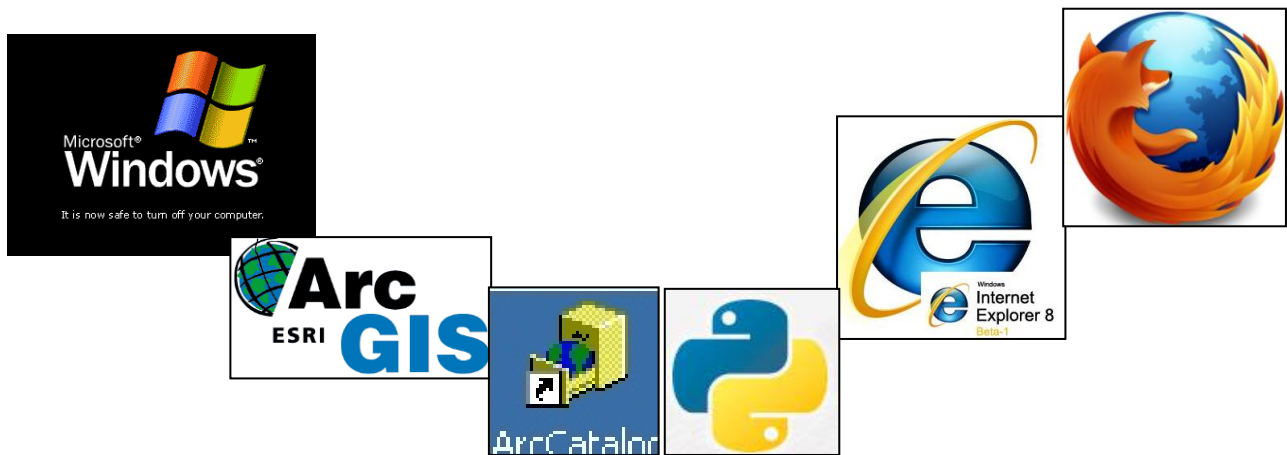


**Figura 26.** Publicación de los metadatos en Geonetwork

### 5.3.2 Software Utilizado

Los requerimientos técnicos son las condiciones informáticas mínimas para el desarrollo del proyecto. A continuación se numeran las tecnologías empleadas necesarias para diseñar, crear, manipula, explotar y mantener la herramienta diseñada para el proyecto.

- Windows xp
- ArcGis
- ArcCatalog
- IDLE (Python)
- Internet Explorer
- Mozilla Firefox



**Figura 27.** Tecnologías empleadas

### **5.3.3 Transformaciones formato XML ISO 19115 a formato ISO 19139**

Dentro de ArcGIS Se proporciona un nuevo traductor de metadatos con la versión actual de ArcGIS for Desktop: el traductor de ArcGIS a ISO 19139, ARCGIS2ISO19139.xml. Todos los estilos de metadatos basados en ISO se han actualizado para utilizar este traductor. Asimismo, cualquier modelo de geoprocésamiento o secuencia de comandos de Python que exporta metadatos al formato ISO 19139 se debe actualizar para utilizar este traductor.

Esri\_ISO2ISO19139.xml —Traduce el contenido almacenado en el formato de metadatos de ArcGIS o el formato de metadatos de Esri-ISO para el formato XML ISO 19139. Este traductor se proporciona para compatibilidad con versiones anteriores para admitir modelos y secuencias de comandos de Python existentes.

## Formato XML:

```
<?xml version="1.0"?>
<metadata xml:lang="es">
  <!-- Esri -->
  <CreateDate>20120309</CreateDate>
  <CreateTime>13011700</CreateTime>
  <ArcGISFormat>1.0</ArcGISFormat>
  <ArcGISStyle>INSPIRE Metadata Directive</ArcGISStyle>
  <SyncOnce>FALSE</SyncOnce>
  <DataProperties>
    <!-- ItemProps -->
    <itemProps>
      <itemName Sync="TRUE">buffer_52x_1_to_10km.tif</itemName>
      <!-- ItemLocation -->
      <itemLocation>
        <linkage Sync="TRUE">file://T:\area_management\coastalBuffer10km-clc-v15\buffer_52x_1_to_10km.tif</linkage>
        <protocol Sync="TRUE">Local Area Network</protocol>
      </itemLocation>
      <imsContentType Sync="TRUE">002</imsContentType>
      <!-- nativeExtBox -->
      <nativeExtBox>
        <westBL Sync="TRUE">1500000.000000</westBL>
        <eastBL Sync="TRUE">7400000.000000</eastBL>
        <southBL Sync="TRUE">900000.000000</southBL>
        <northBL Sync="TRUE">5500000.000000</northBL>
        <extTypeCode Sync="TRUE">1</extTypeCode>
      </nativeExtBox>
      <!-- ItemProps -->
      <!-- coordRef -->
      <coordRef>
        <type Sync="TRUE">Projected</type>
        <geocsn Sync="TRUE">GCS_grs80</geocsn>
        <csUnits Sync="TRUE">Linear Unit: Meter (1.000000)</csUnits>
        <projcsn Sync="TRUE">grs80_Lambert_Azimuthal_Equal_Area</projcsn>
        <pxml Sync="TRUE"><ProjectedCoordinateSystem xsi:type="ProjectedCoordinateSystem" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:typens="http://www.esri.com/schemas/ArcGIS/10.0"><WKT>PROJCS
          [&quot;grs80_Lambert_Azimuthal_Equal_Area&quot;,&quot;GCS_grs80&quot;,&quot;DATUM[&quot;D_ETRS_1989&quot;,&quot;SPHEROID
            [&quot;Geodetic_Reference_System_1980&quot;,&quot;6378137.0,298.257222101]],&quot;PRIMEM[&quot;Greenwich&quot;,&quot;0.0],UNIT
            [&quot;Degree&quot;,&quot;0.0174532925199433]],PROJECTION[&quot;Lambert_Azimuthal_Equal_Area&quot;],PARAMETER
            [&quot;false_easting&quot;,&quot;4321000.0],PARAMETER[&quot;false_northing&quot;,&quot;3210000.0],PARAMETER
            [&quot;central_meridian&quot;,&quot;10.0],PARAMETER[&quot;latitude_of_origin&quot;,&quot;52.0],UNIT[&quot;Meter&quot;,&quot;1.0]]</WKT><XOrigin>
            8426600</XOrigin><YOrigin>9526700</YOrigin><XYScale>353290001.83332509</XYScale><ZOrigin>
            </ZOrigin>
          </ProjectedCoordinateSystem>
        </pxml>
      </coordRef>
    </itemProps>
  </DataProperties>
</metadata>
```

Figura 28. Formato XML

## Formato XML 19139:

```
<?xml version="1.0" encoding="UTF-8"?>
<gmd:MD_Metadata xmlns:gmd="http://www.isotc211.org/2005/gmd" http://www.isotc211.org/2005/gmd/gmd.xsd">
  <gmd:geonets xmlns:geonets="http://www.fao.org/geonetwork" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:gco="http://www.isotc211.org/2005/gco"
  xmlns:gml="http://www.opengis.net/gml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:gmd="http://www.isotc211.org/2005/gmd">
    <!-- gmd:fileIdentifier -->
    <gmd:fileIdentifier>
      <gco:CharacterString>f68623ca-c1ff-44b6-a582-1487f43fc93</gco:CharacterString>
    </gmd:fileIdentifier>
    <!-- gmd:language -->
    <gmd:language>
      <gmd:LanguageCode codeListValue="eng" codeList="http://www.loc.gov/standards/iso639-2/">eng</gmd:LanguageCode>
    </gmd:language>
    <!-- gmd:characterSet -->
    <gmd:characterSet>
      <gmd:MD_CharacterSetCode codeListValue="utf8"
      codeList="http://standards.iso.org/ittf/PubliclyAvailableStandards/ISO_19139_Schemas/resources/Codelist/ML_gmxCodetlists.xml#MD_CharacterSetCode"/>
    </gmd:characterSet>
    <!-- gmd:hierarchylevels -->
    <gmd:MD_ScopeCode codeListValue="dataset"
    codeList="http://standards.iso.org/ittf/PubliclyAvailableStandards/ISO_19139_Schemas/resources/Codelist/ML_gmxCodetlists.xml#MD_ScopeCode"/>
    </gmd:hierarchylevels>
    <!-- gmd:contact -->
    <gmd:contact>
      <gmd:CI_Responsibility>
        <gmd:organisationName>
          <gco:CharacterString>European Environment Agency</gco:CharacterString>
        </gmd:organisationName>
        <gmd:contactInfo>
          <gmd:CI_Contact>
            <gmd:address>
              <gmd:CI_Address>
                <gmd:deliveryPoint>
                  <gco:CharacterString>Kongens Nytorv 6</gco:CharacterString>
                </gmd:deliveryPoint>
                <gmd:city>
                  <gco:CharacterString>Copenhagen</gco:CharacterString>
                </gmd:city>
                <gmd:administrativeArea>
                  <gco:CharacterString>K</gco:CharacterString>
                </gmd:administrativeArea>
                <gmd:postalCode>
                  </gmd:postalCode>
                </gmd:CI_Address>
              </gmd:address>
            </gmd:CI_Contact>
          </gmd:contactInfo>
        </gmd:CI_Contact>
      </gmd:CI_Responsibility>
    </gmd:contact>
  </gmd:geonets>
</gmd:MD_Metadata>
```

Figura 29. Formato XML 19139

Recorrido de las carpetas donde se encuentra almacenados los metadatos y la exportación a formato ISO XML 19139.

```

setProjection.py - Y:\Personal\m_lozano\setProjection.py
File Edit Format Run Options Windows Help

#Import modules
import arcpy, sys, os
from arcpy import env

# --- setProjection.py ---
# Sets the projection for all the layers contained in a folder
# AND SUBFOLDERS (take care!!!).
# It works for both raster and vector layers.
# Params:
# inWorkspace: the base folder to apply the script
# [projectionCode] the code of the projection to apply. By default,
# code 3035 (ETRS89 LAEA) is applied
# -----

def setProjection(folder, crsCode=3035, recurseDirectories=True):
    logging.info("** Processing folder: " + folder + "...")
    folderbak = env.workspace
    env.workspace = folder
    doSetProjection(folder, crsCode)
    if (recurseDirectories):
        wplst = arcpy.ListWorkspaces("", "ALL")
        for wp in wplst:
            setProjection(wp, crsCode, recurseDirectories)
    env.workspace = folderbak

def doSetProjection(folder, crsCode):
    folderbak = env.workspace
    env.workspace = folder
    datasets = arcpy.ListDatasets("")
    for dataset in datasets:
        # print dataset
        logging.info("** Processing dataset: " + dataset + "...")
        dir = arcpy.GetInstallInfo("desktop")["InstallDir"]
        translator = dir + "Metadata/Translator/ArcGIS2ISO19139.xml"
        arcpy.ExportMetadata_conversion (dataset, translator, dataset+".iso.xml")

        print env.workspace
        print dataset
        #arcpy.DefineProjection_management (dataset, crsCode)
        #arcpy.ExportRasterWorldFile_management (dataset)
    env.workspace = folderbak

```

Figura 30. Script en Python recorre cada una de las carpetas y exporta

## 6. DESARROLLO DE LA HERRAMIENTA

Aquí se mostrara cada uno de los pasos que se realizaron para el desarrollo del diseño para la edición y publicación de metadatos, basado en la integración de ArcGis y Geonetwork. A continuación se muestra un esquema de los pasos a seguir.





## 6.1 Producción y actualización de metadatos (Metadata Editor)

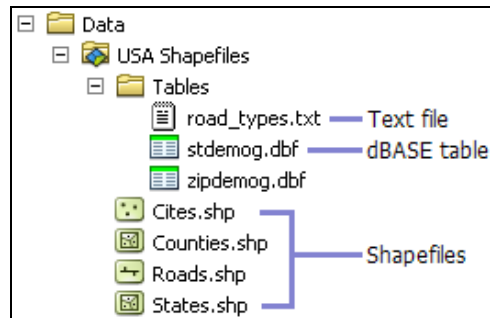
Dentro del mundo GIS existen diferentes contenedores de información geográfica como son los raster y los vectoriales. Por otro lado, las pocas herramientas que existen dentro de GIS ofrecen una deducción automática de metadatos para los contenedores raster y vectoriales, se basan en el análisis de estos formatos específicos y la aplicación de mecanismos *ad-hoc* los cuales procesan los datos para extraer la información que sea utilizada en la construcción de los elementos de metadatos (**Oliver S, et al** 2001).

Herramientas como CatMDEdi es utilizada para realizar una extracción automática de los metadatos en diferentes formatos. De esta manera existen elementos que solo pueden ser extraídos de cierto tipo de datos y archivos, mientras que por ejemplo el tamaño de los datos pueden obtenerse en cualquier circunstancia (**Oliver S, et al** 2001). Esta herramienta también presenta algunos inconvenientes como por ejemplo no hay una relación directa de los datos con los metadatos. Debido a la falta de relación directa hace que no se pueda extraer cierta información automática de los datos como los son las coordenadas, el sistema de referencia, sistema de coordenadas, etc. El CatMDEdit resulta una herramienta complicada en cuanto a la diversidad de campos que hay para completar, mientras que el editor de ArcCatalog contenía menos campos de información, y además no mostraba la información que extraía directamente de los datos (**Carbajal J, 2007**).

Otra herramienta muy utilizada y conocida para la generación automática de metadatos de datos geográficos es ESRI ArcCatalog, esta herramienta permite la carga automática de un número de campos básicos y la actualización de la sincronización de datos y metadatos. Para mejorar el desarrollo de la herramienta se han creado algunas extensiones, tales como el editor de metadatos de NEM (Núcleo Español de Metadatos), herramienta integrada a ArcCatalog capaz de generar un registro que cumple con el estándar ISO 19115:2003 y NEM (**Oliver S, et al** 2001).

Los datos se encuentran almacenados en diversos formatos como Shapefile, Personal Geodatabase, Geodatabase de Archivos, GeoTiff, Grid.

Shapefile: Es un formato sencillo no topológico en el que se almacena la ubicación geométrica y la información de atributos de las entidades geográficas. Las entidades geográficas en un shapefile se representan por medio de puntos, líneas o polígonos. Los shapefile también pueden incluir tablas de dBASE, las cuales pueden almacenar atributos adicionales que se vinculan a las entidades de un shapefile (<http://www.resources.arcgis.com>).



Personal Geodatabase: Base de datos de Microsoft Access donde se puede almacenar, consultar y administrar tanto datos espaciales como no espaciales. Las geodatabases personales tienen un tamaño máximo de 2 GB. Solo una persona puede editar a la vez los datos de una geodatabase personal (<http://www.resources.arcgis.com>).

Geodatabase de Archivos: Una geodatabase de archivos es una colección de archivos en una carpeta en el disco que puede almacenar, consultar y administrar datos espaciales y datos no espaciales (<http://www.resources.arcgis.com>).

En la siguiente **Figura 31**, se observa los pasos para la producción y actualización de metadatos que se llevó a cabo durante el proyecto.

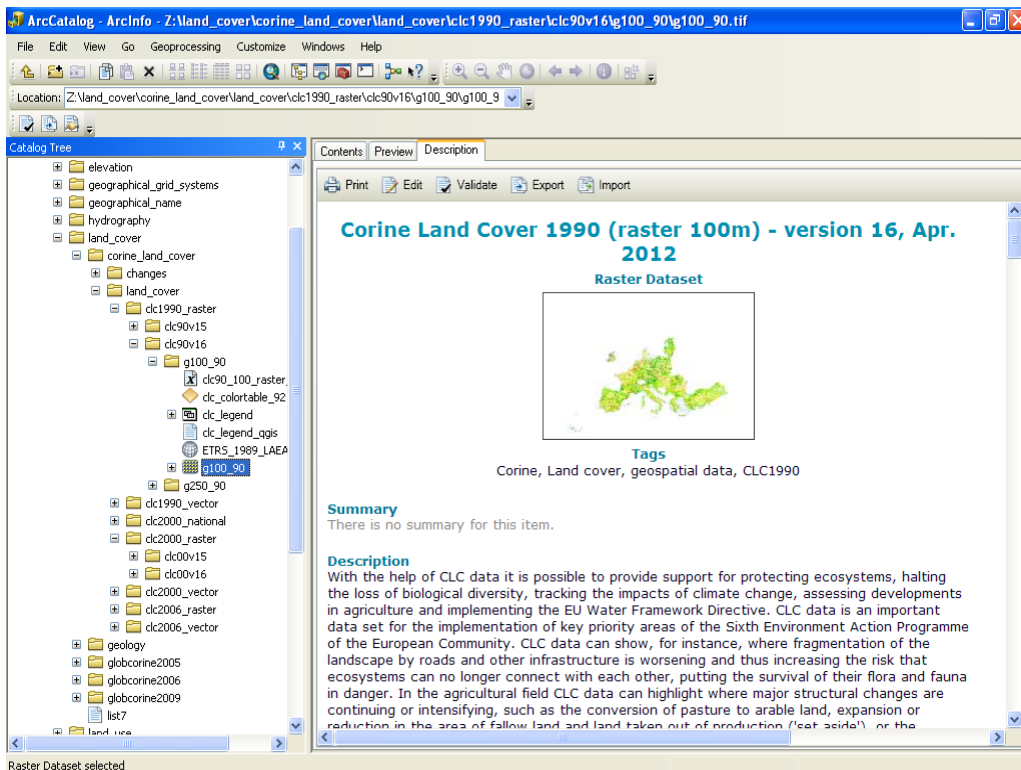
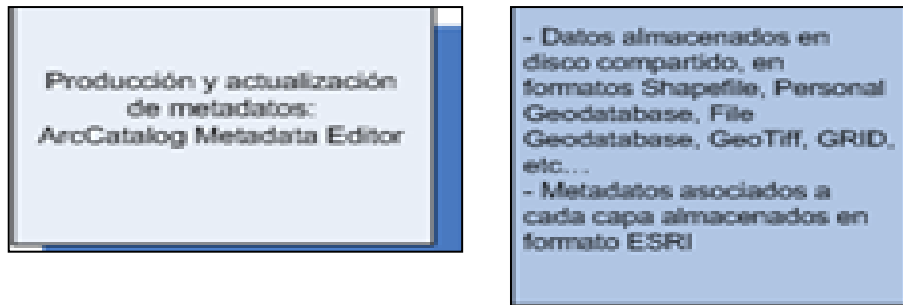


Figura 31. Producción y generación de metadatos en ArcCatalog

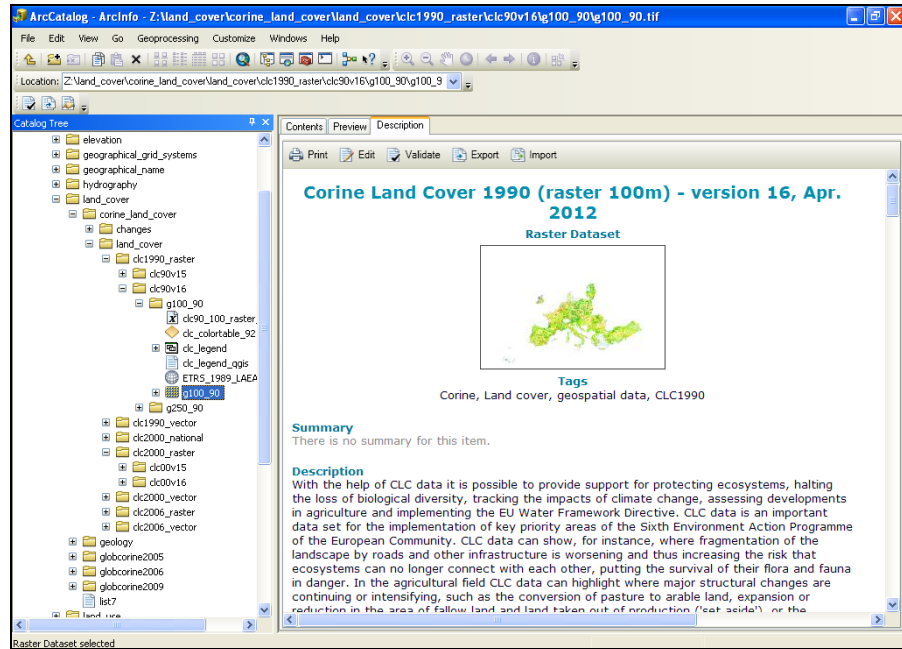
## 6.2 Exportación a XML

Para llevar a cabo la exportación a XML lo primero que se debe realizar es la actualización de los metadatos para que contengan las propiedades más recientes del elemento ArcGIS antes de procesar los metadatos y finalmente los exporta a un archivo XML que se ajusta al formato de metadatos estándar.

El procedimiento anterior se realiza con el fin de producir los mejores resultados en un formato estándar. Después los metadatos se exportan mediante el *Traductor de metadatos*. Se proporciona un nuevo traductor de metadatos con la versión actual de ArcGIS for Desktop: el traductor de ArcGIS a ISO 19139, ARCGIS2ISO19139.xml. Cualquier modelo de geoprociamiento o secuencia de comandos de Python que exporta metadatos al formato ISO 19139 se debe actualizar para utilizar este traductor (<http://www.resources.arcgis.com>).

Los metadatos producido en la versión 9.3.1 existentes en el formato ESRI-ISO se deben actualizar al formato de metadatos de ArcGIS para poder exportar correctamente el contenido de los metadatos del elemento al formato ISO 19139 con el traductor de ArcGIS a ISO 19139, ARCGIS2ISO19139. Xml (<http://www.resources.arcgis.com>).

En el script de Python vemos como aactualiza y exporta los diferentes metadatos ArcGIS encontrados en cada una de las carpetas, a un archivo XML que tiene el formato correcto para el estándar de metadatos ISO 19139.



```

setProjection.py - Y:\Personal\m_lozano\setProjection.py
File Edit Format Run Options Windows Help

def doSetProjection(folder, crsCode):
    folderbak = env.workspace
    env.workspace = folder
    datasets = arcpy.ListDatasets("")
    for dataset in datasets:
        # print dataset
        logging.info("** Processing dataset: " + dataset + "...")
        dir = arcpy.GetInstallInfo("desktop")["InstallDir"]
        translator = dir + "Metadata/Translator/ArcGIS218019139.xml"
        arcpy.ExportMetadata_conversion (dataset, translator, dataset+".iso.xml")

        print env.workspace
        print dataset
        #arcpy.DefineProjection_management (dataset, crsCode)
        #arcpy.ExportRasterWorldFile_management (dataset)
    env.workspace = folderbak

import sys, time, logging
import traceback

def main():
    try:
        inputFolder = "Z:\\TESTLIGIT"
        setProjection(inputFolder)
    except:
        exc, value, tcb = sys.exc_info()
        logging.error(str(exc) + "\n" + str(value))
        outlog = open(os.path.expanduser("~/")+"/etc/siatb'+pid+'.txt', 'w')
        traceback.print_tb(tcb, 10, outlog)
        outlog.close()
        raise

if __name__ == "__main__":
    main()

```

Figura 32. Exportación a XML

### 6.3 Metadatos exportados a formato XML ISO 19139

Para realizar la exportación de metadatos, primero se deben actualizar para incluir las últimas propiedades del elemento de ArcGis, que describe antes de transformar la información en un archivo XML que usa un formato de metadatos estándar.

```
Script_final_17Dic.py - G:\Proyecto Final LIGIT\Versiones script final\Script_final_17Dic.py
File Edit Format Run Options Windows Help

direct = arcpy.GetInstallInfo("desktop") ["InstallDir"]
translator = direct + "Metadata/Translator/ArcGIS2ISO19139.xml"
folderbak = env.workspace
env.workspace = folder
fcList = arcpy.ListFeatureClasses()
for fc in fcList:
    desc = arcpy.Describe(fc)
    print folderbak+ "\\\" + fc + ".iso.xml"
    if os.path.exists(folderbak+ "/" + fc + ".iso.xml"):
        os.remove(folderbak + "/" + fc + ".iso.xml")
    #logging.info(" ** Processing fc: " + fc + "...")
    try:
        arcpy.ExportMetadata_conversion (fc, translator, folderbak+"/"+ fc + ".iso.xml")
        insertDataPath(folderbak+ "\\\" + fc, folderbak+ "\\\" + fc + ".iso.xml")
    except:
        exc,value,tcb = sys.exc_info()
        print str(exc) + "\n" + str(value)
        traceback.print_tb(tcb, 10)
    print "FC"
    print "fc " + fc
    print "FIN-----"

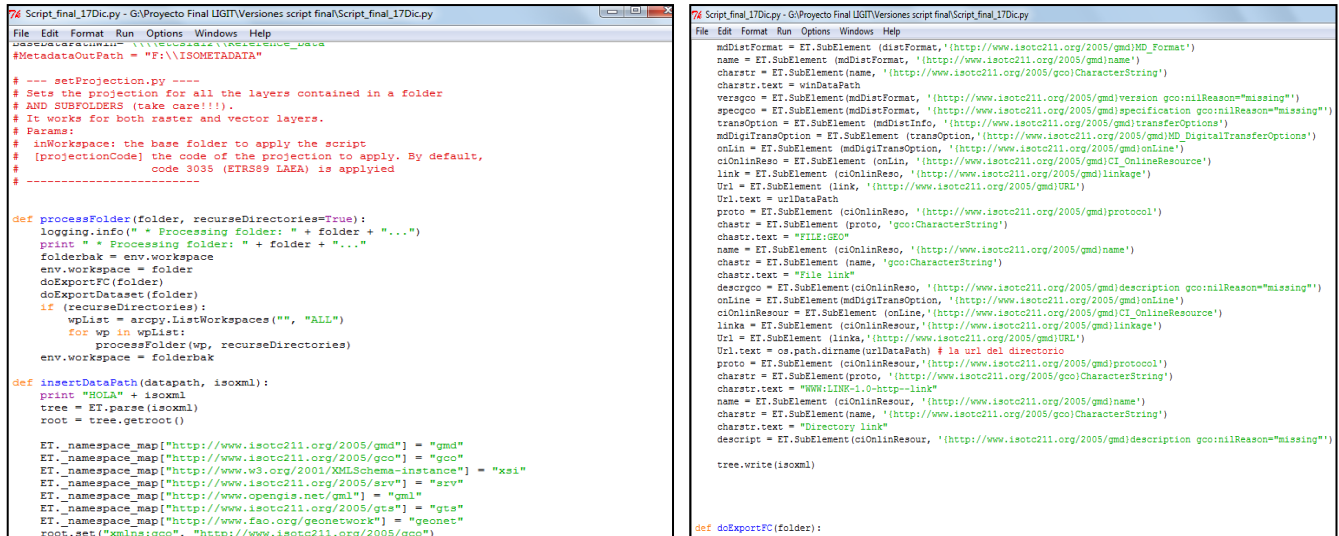
folderbak = env.workspace
```



```
<?xml version="1.0" encoding="UTF-8"?>
<gmd:MD_Metadata xmlns:gmd="http://www.isotc211.org/2005/gmd" xmlns:gco="http://www.isotc211.org/2005/gco/gmd/gco"
xmlns:geonet="http://www.fao.org/geonetwork" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:gco="http://www.isotc211.org/2005/gco"
xmlns:gml="http://www.opengis.net/gml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:gmd="http://www.isotc211.org/2005/gmd">
  <gmd:fileIdentifier>
    <gco:CharacterString>f68623ca-c1ff-44b6-a582-14872f43fc93</gco:CharacterString>
  </gmd:fileIdentifier>
  <gmd:language>
    <gmd:LanguageCode codeListValue="eng" codeList="http://www.loc.gov/standards/iso639-2/">
    </gmd:language>
  <gmd:characterSet>
    <gmd:MD_CharacterSetCode codeListValue="utf8"
codeList="http://standards.iso.org/ittf/PubliclyAvailableStandards/ISO_19139_Schemas/resources/Codelist/ML_gmxCodeLists.xml#MD_CharacterSetCode"/>
  </gmd:characterSet>
  <gmd:hierarchyLevel>
    <gmd:MD_ScopeCode codeListValue="dataset"
codeList="http://standards.iso.org/ittf/PubliclyAvailableStandards/ISO_19139_Schemas/resources/Codelist/ML_gmxCodeLists.xml#MD_ScopeCode"/>
  </gmd:hierarchyLevel>
  <gmd:contact>
    <gmd:CI_ResponsibleParty>
      <gmd:organisationName>
        <gco:CharacterString>European Environment Agency</gco:CharacterString>
      </gmd:organisationName>
      <gmd:contactInfo>
        <gmd:CI_Contact>
          <gmd:address>
            <gmd:CI_Address>
              <gmd:deliveryPoint>
                <gco:CharacterString>Kongens Nytorv 6</gco:CharacterString>
              </gmd:deliveryPoint>
              <gmd:city>
                <gco:CharacterString>Copenhagen</gco:CharacterString>
              </gmd:city>
              <gmd:administrativeArea>
                <gco:CharacterString>K</gco:CharacterString>
              </gmd:administrativeArea>
              <gmd:postalCode>
```

Figura 33. Exportación a formato XML ISO 19139

## 6.4 Importación al catálogo de Geonetwork



```
Script_final_17Dic.py - G:\Proyecto Final UGIT\Versiones script final\Script_final_17Dic.py
File Edit Format Run Options Windows Help
#-----
#MetadataOutPath = "F:\ISOMETADATA"
#-----
# --- setProjection.py ---
# Sets the projection for all the layers contained in a folder
# AND SUBFOLDERS (take care!!!).
# It works for both raster and vector layers.
# Params:
# inWorkspace: the base folder to apply the script
# [projectionCode] the code of the projection to apply. By default,
# code 3035 (ETRS89 LAEA) is applied
#-----

def processFolder(folder, recurseDirectories=True):
    logging.info(" * Processing folder: " + folder + "...")
    print " * Processing folder: " + folder + "..."
    folderbak = env.workspace
    env.workspace = folder
    doExportFC(folder)
    doExportDataset(folder)
    if (recurseDirectories):
        wplist = arcpy.ListWorkspaces("", "ALL")
        for wp in wplist:
            processFolder(wp, recurseDirectories)
    env.workspace = folderbak

def insertDataPath(datapath, isoxml):
    print "HOJA" + isoxml
    tree = ET.parse(isoxml)
    root = tree.getroot()

    ET_namespace_map["http://www.isotc211.org/2005/gmd"] = "gmd"
    ET_namespace_map["http://www.isotc211.org/2005/gco"] = "gco"
    ET_namespace_map["http://www.w3.org/2001/XMLSchema-instance"] = "xsi"
    ET_namespace_map["http://www.isotc211.org/2005/srv"] = "srv"
    ET_namespace_map["http://www.opengis.net/gml"] = "gml"
    ET_namespace_map["http://www.isotc211.org/2005/gts"] = "gts"
    ET_namespace_map["http://www.fao.org/geonetwork"] = "geonet"
    root.set("xmlns:geonet", "http://www.isotc211.org/2005/gco")

    mdDistFormat = ET.SubElement(distFormat, '{http://www.isotc211.org/2005/gmd}MD_Format')
    name = ET.SubElement(mdDistFormat, '{http://www.isotc211.org/2005/gmd}name')
    charstr = ET.SubElement(name, '{http://www.isotc211.org/2005/gco}CharacterString')
    charstr.text = winDataPath

    versgco = ET.SubElement(mdDistFormat, '{http://www.isotc211.org/2005/gmd}version gco:nilReason="missing"')
    specgco = ET.SubElement(mdDistFormat, '{http://www.isotc211.org/2005/gmd}specification gco:nilReason="missing"')
    transOption = ET.SubElement(mdDistInfo, '{http://www.isotc211.org/2005/gmd}transferOptions')
    mdDigiTransOption = ET.SubElement(transOption, '{http://www.isotc211.org/2005/gmd}MD_DigitalTransferOptions')
    onLin = ET.SubElement(mdDigiTransOption, '{http://www.isotc211.org/2005/gmd}onLine')
    ciOnlinReso = ET.SubElement(onLin, '{http://www.isotc211.org/2005/gmd}CI_OnlineResource')
    link = ET.SubElement(ciOnlinReso, '{http://www.isotc211.org/2005/gmd}linkage')
    url = ET.SubElement(link, '{http://www.isotc211.org/2005/gmd}URL')
    url.text = urlDataPath

    proto = ET.SubElement(ciOnlinReso, '{http://www.isotc211.org/2005/gmd}protocol')
    charstr = ET.SubElement(proto, '{http://www.isotc211.org/2005/gmd}name')
    charstr.text = "FILE:GEO"

    name = ET.SubElement(ciOnlinReso, '{http://www.isotc211.org/2005/gmd}name')
    charstr = ET.SubElement(name, '{http://www.isotc211.org/2005/gmd}name')
    charstr.text = "File link"

    descrqco = ET.SubElement(ciOnlinReso, '{http://www.isotc211.org/2005/gmd}description gco:nilReason="missing"')
    onLine = ET.SubElement(mdDigiTransOption, '{http://www.isotc211.org/2005/gmd}onLine')
    ciOnlinResour = ET.SubElement(onLine, '{http://www.isotc211.org/2005/gmd}CI_OnlineResource')
    links = ET.SubElement(ciOnlinResour, '{http://www.isotc211.org/2005/gmd}linkage')
    url = ET.SubElement(links, '{http://www.isotc211.org/2005/gmd}URL')
    url.text = os.path.dirname(urlDataPath) # la url del directorio
    proto = ET.SubElement(ciOnlinResour, '{http://www.isotc211.org/2005/gmd}protocol')
    charstr = ET.SubElement(proto, '{http://www.isotc211.org/2005/gco}CharacterString')
    charstr.text = "WWW:LINK-1.0-http--link"
    name = ET.SubElement(ciOnlinResour, '{http://www.isotc211.org/2005/gmd}name')
    charstr = ET.SubElement(name, '{http://www.isotc211.org/2005/gmd}name')
    charstr.text = "Directory link"
    descrpt = ET.SubElement(ciOnlinResour, '{http://www.isotc211.org/2005/gmd}description gco:nilReason="missing"')

    tree.write(isoxml)

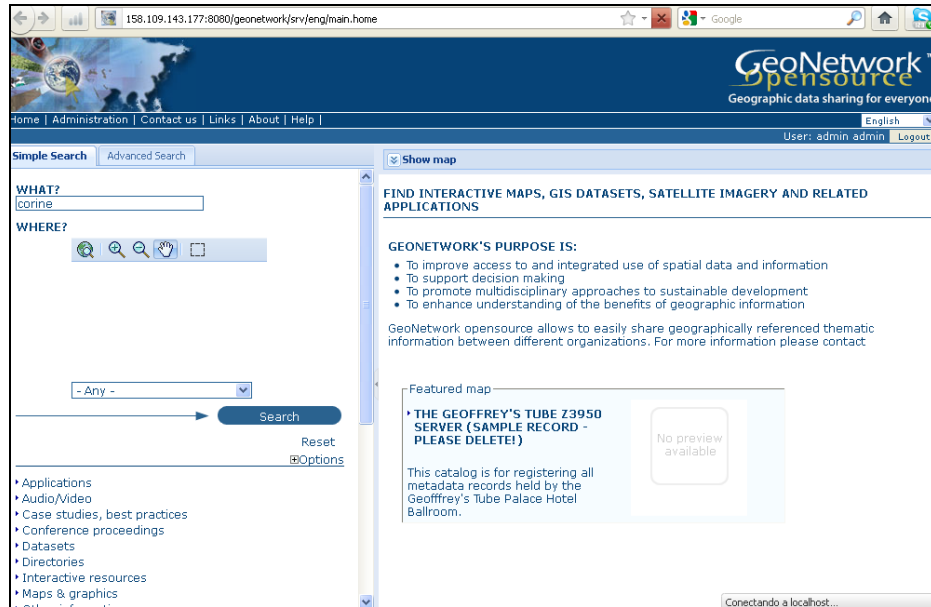
def doExportFC(folder):
```

Figura 34. Importación al catálogo de Geonetwork

## 6.5 Catálogo de metadatos Geonetwork

En el catálogo de metadatos en Geonetwork se realizaron los siguientes pasos:

1. Acceder al portal con el nombre de usuario y contraseña (inicio de sesión), posterior visualización del portal de Geonetwork.
  2. Recolecta de metadatos (Harvesting)
  3. Comprobación del Harvesting hecho en los metadatos
  4. Comprobación de adición de los metadatos
  5. Visualización de los metadatos publicados (adición)
  6. Modificación de Geonetwork para la ruta a los datos
- 
1. Primero se lleva a cabo el inicio de sesión al portal de Geonetwork, posterior se accede al portal donde se puede realizar las diferentes búsquedas de metadatos, allí aparecerán aquellos metadatos que han sido publicados.



**Figura 35.** Acceso al portal de Geonetwork

2. Se realiza la recolecta de metadatos (harvesting). Los metadatos exportados a Geonetwork, deben estar de forma estandarizada evitando que importantes partes del conocimiento de los datos se pierdan. Los estandares de metadatos soportados por Geonetworkusan la implementación del esquema XML para metadatos de información geografica, especificación tecnica 191139 de ISO para la codificación XML.

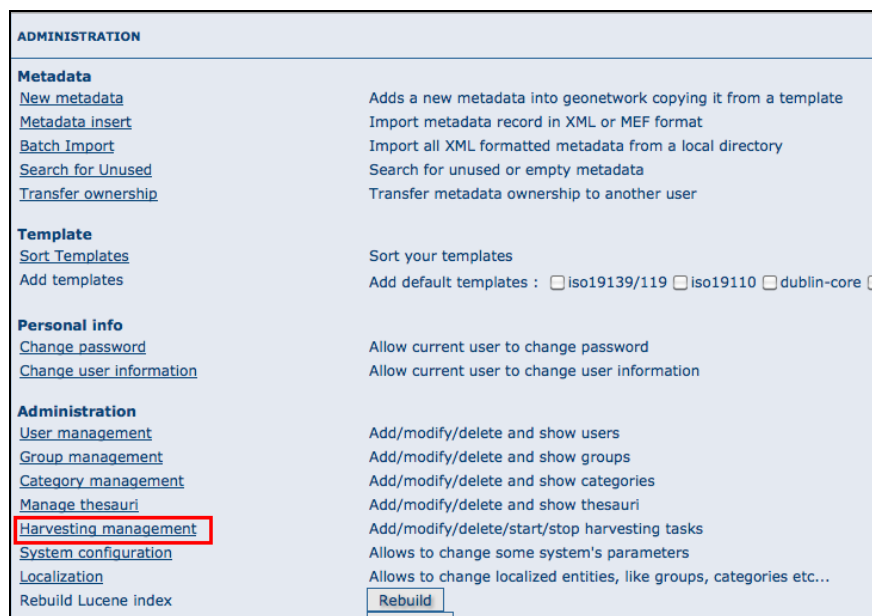
Geonetwork soporta diversos perfiles de metadatos; estos perfiles pueden adoptar la forma de plantillas las cuales se pueden crear usando el editor de metadatos.

Siendo el estandar de metadatos ISO 19115:2003 para información geografica el utilizado y preferido para muchos, se ha tenido que migrar los metadatos antiguos al nuevo estándar.



Esta capacidad de Geonetwork nos permite obtener metadatos de otros catálogos creando así una jerarquía de catálogos.

➤ Administration -> Harvesting management



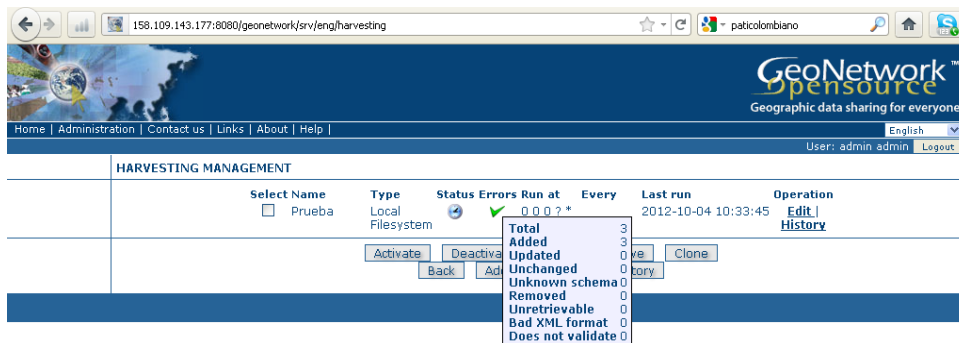
**Figura 36.** Recolección de metadatos (harvesting) en carpetas locales

3. Posteriormente se observa si al adicionar los metadatos se realizó de forma correcta o por el contrario se produjo algún error, en la siguiente figura se observan ambos casos.



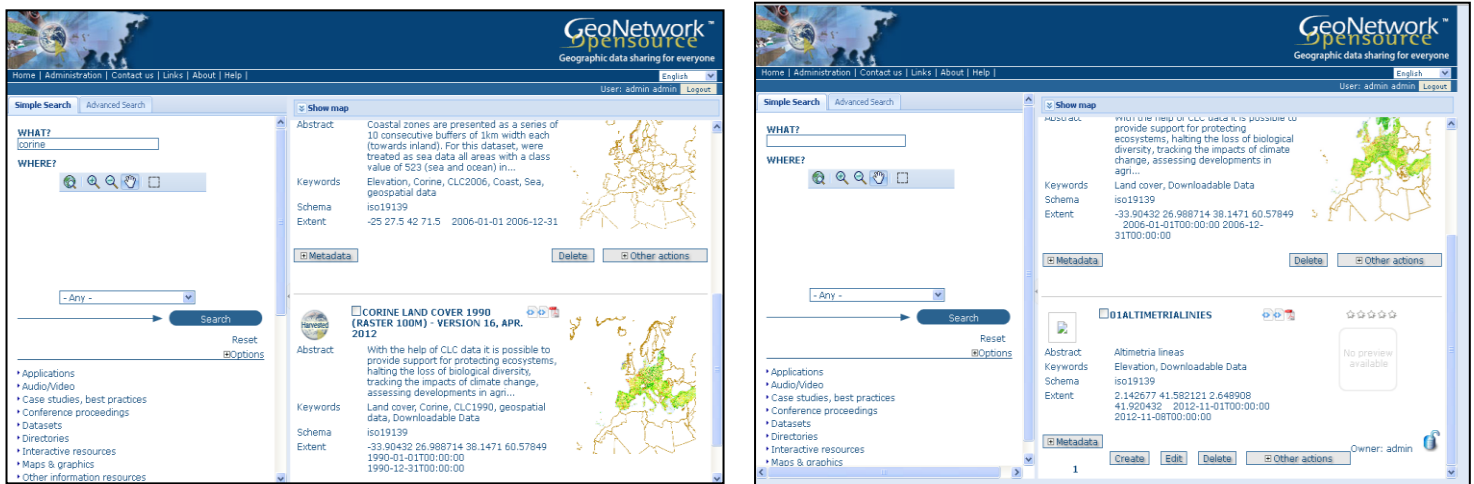
**Figura 37.** Visualización de adición de metadatos en Geonetwork

- En este punto se observa cuantos metadatos fueron adicionados en el catálogo de Geonetwork. Como se muestra en la Figura aparecen el total de metadatos adicionados pero también aparece la palabra adición indicado de igual manera el número que adiciono.



**Figura 38.** Numero de metadatos adicionados a Geonetwork

- En este punto despues de habaer realizado la edición, podemos regresar a la pagina principal de Geonetwork, y alli hacemos una busqueda donde encontramos publicados cada uno de los metadatos que han sido añadidos correctamente al catalogo.

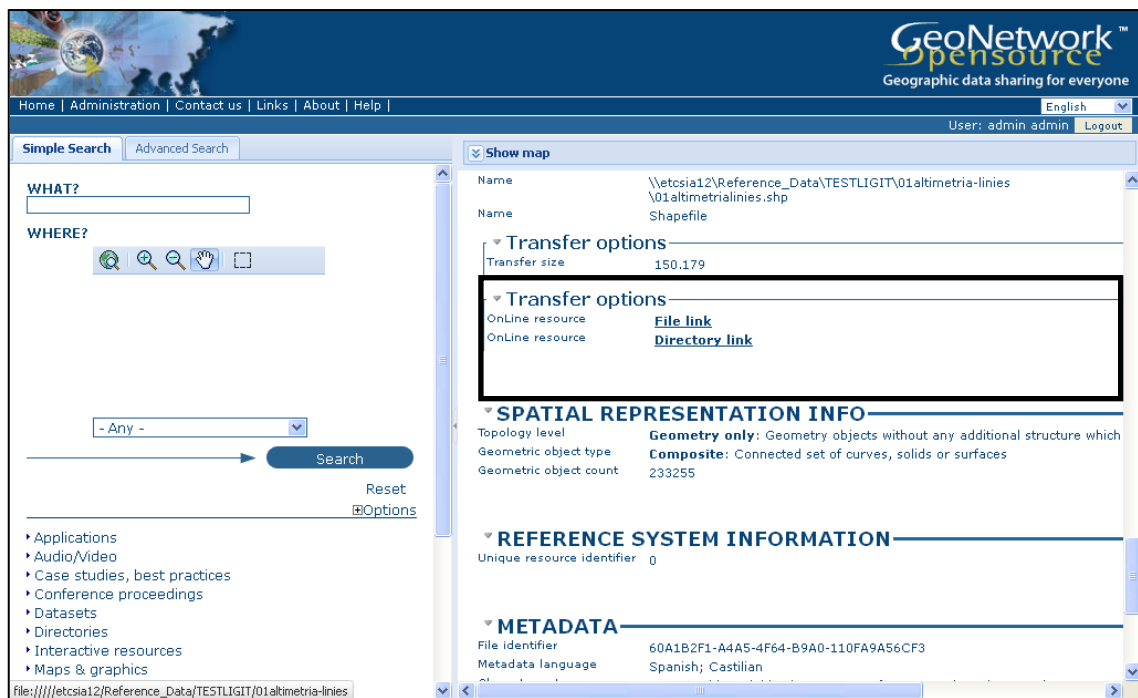


**Figura 39.** Publicación de metadatos en el portal Geonetwork

6. Después de visualizar los metadatos en el catálogo, se procede a la modificación de Geonetwork para la ruta a los datos siendo este el objetivo final del proyecto. Este procedimiento se llevó a cabo en 3 fases.

- Visualización para saber dónde se va a insertar la ruta a los datos directamente desde el catalogo (Geonetwork).
- Configuración mediante un script (Python) para acceder a los datos directamente desde el catalogo.
- Configuración de Internet Exploret8 y Mozilla Firefox, para visualizar de forma correcta la ruta a los datos desde Geonetwork.

En la primera fase se realiza la visualización de donde se va insertar la ruta para el acceso directo a los datos. El campo señalado en rojo es el que fue modificado para acceder directamente a los datos.



**Figura 40.** Campo de visualización para insertar la ruta directa a los datos

En la segunda fase se procede a la configuración de Geonetwork, esta fase se realiza mediante un script.

```

Script_final_17Dic.py - G:\Proyecto Final LGIT\Versiones script final\Script_final_17Dic.py
File Edit Format Run Options Windows Help
#-----
#MetadataOutPath = "F:\ISOMETADATA"

# --- setProjection.py ---
# Sets the projection for all the layers contained in a folder
# AND SUBFOLDERS (take care!!!).
# It works for both raster and vector layers.
# Params:
# inWorkspace: the base folder to apply the script
# [projectionCode] the code of the projection to apply. By default,
# code 3035 (ETRS89 LAEA) is applied
# -----

def processFolder(folder, recurseDirectories=True):
    logging.info(" * Processing folder: " + folder + "...")
    print " * Processing folder: " + folder + "..."
    folderbak = env.workspace
    env.workspace = folder
    doExportFC(folder)
    doExportDataset(folder)
    if (recurseDirectories):
        wpList = arcpy.ListWorkspaces("", "ALL")
        for wp in wpList:
            processFolder(wp, recurseDirectories)
    env.workspace = folderbak

def insertDataPath(datapath, isoxml):
    print "HOLA" + isoxml
    tree = ET.parse(isoxml)
    root = tree.getroot()

    ET._namespace_map["http://www.isotc211.org/2005/gmd"] = "gmd"
    ET._namespace_map["http://www.isotc211.org/2005/gco"] = "gco"
    ET._namespace_map["http://www.w3.org/2001/XMLSchema-instance"] = "xsi"
    ET._namespace_map["http://www.isotc211.org/2005/srv"] = "srv"
    ET._namespace_map["http://www.opengis.net/gml"] = "gml"
    ET._namespace_map["http://www.isotc211.org/2005/gts"] = "gts"
    ET._namespace_map["http://www.fao.org/geonetwork"] = "geonet"
    root.set("xmlns:gco", "http://www.isotc211.org/2005/gco")

mdDistFormat = ET.SubElement (distFormat, '{http://www.isotc211.org/2005/gmd}MD_Format')
name = ET.SubElement (mdDistFormat, '{http://www.isotc211.org/2005/gmd}name')
charstr = ET.SubElement (name, '{http://www.isotc211.org/2005/gco}CharacterString')
charstr.text = winDataPath
versgco = ET.SubElement (mdDistFormat, '{http://www.isotc211.org/2005/gmd}version gco:nilReason="missing"')
specgco = ET.SubElement (mdDistFormat, '{http://www.isotc211.org/2005/gmd}specification gco:nilReason="missing"')
transOption = ET.SubElement (mdDistInfo, '{http://www.isotc211.org/2005/gmd}transferOptions')
mdDigiTransOption = ET.SubElement (transOption, '{http://www.isotc211.org/2005/gmd}MD_DigitalTransferOptions')
onLin = ET.SubElement (mdDigiTransOption, '{http://www.isotc211.org/2005/gmd}onLine')
ciOnLinReso = ET.SubElement (onLin, '{http://www.isotc211.org/2005/gmd}CI_OnlineResource')
link = ET.SubElement (ciOnLinReso, '{http://www.isotc211.org/2005/gmd}linkage')
url = ET.SubElement (link, '{http://www.isotc211.org/2005/gmd}URL')
url.text = urlDataPath
proto = ET.SubElement (ciOnLinReso, '{http://www.isotc211.org/2005/gmd}protocol')
charstr = ET.SubElement (proto, 'gco:CharacterString')
name = ET.SubElement (ciOnLinReso, '{http://www.isotc211.org/2005/gmd}name')
charstr = ET.SubElement (name, 'gco:CharacterString')
charstr.text = "File link"
descrgco = ET.SubElement (ciOnLinReso, '{http://www.isotc211.org/2005/gmd}description gco:nilReason="missing"')
onLine = ET.SubElement (mdDigiTransOption, '{http://www.isotc211.org/2005/gmd}onLine')
ciOnLinResour = ET.SubElement (onLine, '{http://www.isotc211.org/2005/gmd}CI_OnlineResource')
links = ET.SubElement (ciOnLinResour, '{http://www.isotc211.org/2005/gmd}linkage')
url = ET.SubElement (links, '{http://www.isotc211.org/2005/gmd}URL')
url.text = os.path.dirname(urlDataPath) # la url del directorio
proto = ET.SubElement (ciOnLinResour, '{http://www.isotc211.org/2005/gmd}protocol')
charstr = ET.SubElement (proto, '{http://www.isotc211.org/2005/gco}CharacterString')
charstr.text = "WWW:LINK-1.0-http--link"
name = ET.SubElement (ciOnLinResour, '{http://www.isotc211.org/2005/gmd}name')
charstr = ET.SubElement (name, '{http://www.isotc211.org/2005/gco}CharacterString')
charstr.text = "Directory link"
descript = ET.SubElement (ciOnLinResour, '{http://www.isotc211.org/2005/gmd}description gco:nilReason="missing"')

tree.write(isoxml)

def doExportFC(folder):

```

```

</gmd:northBoundLatitude>
</gmd:EX_GeographicBoundingBox>
</gmd:geographicElement>
</gmd:EX_Extent>
</gmd:extent>
- <gmd:supplementalInformation gco:nilReason="missing">
  <gco:CharacterString/>
</gmd:supplementalInformation>
</gmd:MD_DataIdentification>
</gmd:identificationInfo>
- <gmd:distributionInfo>
  - <gmd:MD_Distribution>
    - <gmd:distributionFormat>
      - <gmd:MD_Format>
        - <gmd:name>
          <gco:CharacterString>Z:\TESTLIGIT\area_management\buffer_523_1_to_10km.tif</gco:CharacterString>
        </gmd:name>
      - <gmd:version gco:nilReason="missing">
        <gco:CharacterString/>
      </gmd:version>
      - <gmd:specification gco:nilReason="missing">
        <gco:CharacterString/>
      </gmd:specification>
    </gmd:MD_Format>
  </gmd:distributionFormat>
- <gmd:transferOptions>
  - <gmd:MD_DigitalTransferOptions>
    - <gmd:onLine>
      - <gmd:CI_OnlineResource>
        - <gmd:linkage>
          <gmd:URL>file:///Z:\TESTLIGIT\area_management</gmd:URL>
          </gmd:linkage>
        - <gmd:protocol>
          <gco:CharacterString>WWW:LINK-1.0-http--link</gco:CharacterString>
        </gmd:protocol>
        - <gmd:name>
          <gco:CharacterString>Z:\TESTLIGIT\area_management</gco:CharacterString>
        </gmd:name>
      - <gmd:description gco:nilReason="missing">

```

Figura 41. Script para generar la ruta a los datos

Al visualizar los metadatos en Geonetwork ya aparecerá la ruta de donde se accederá a los datos.

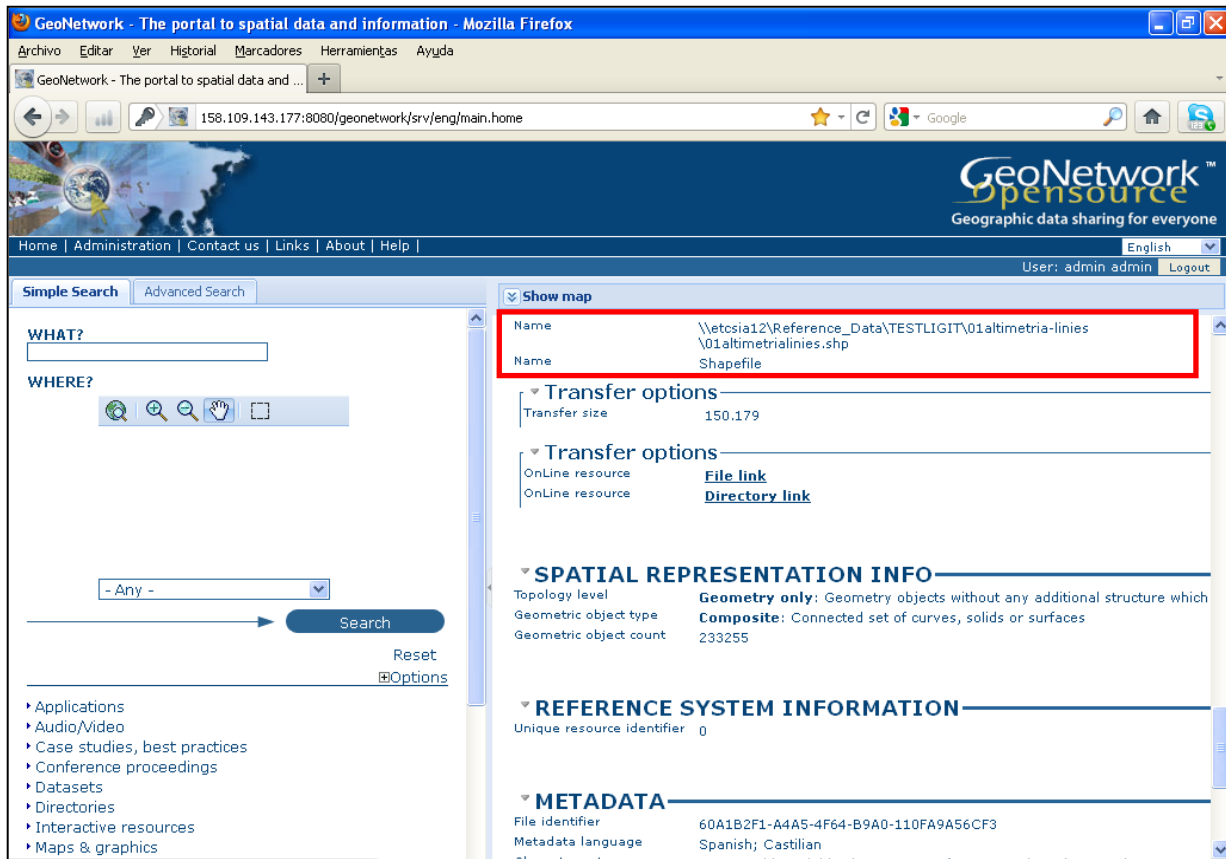


Figura 42. Visualización de la ruta en el catálogo de Geonetwork

En la tercera fase ya una vez realizado la configuración de la ruta procedemos a configurar Internet Explores8 y Mozilla Firefox.

#### Internet Explorer 8:

Los pasos a seguir para la configuración de Internet Explorer fueron; **Herramientas- Opciones de internet- seguridad-intranet local- sitios-opciones avanzadas.**

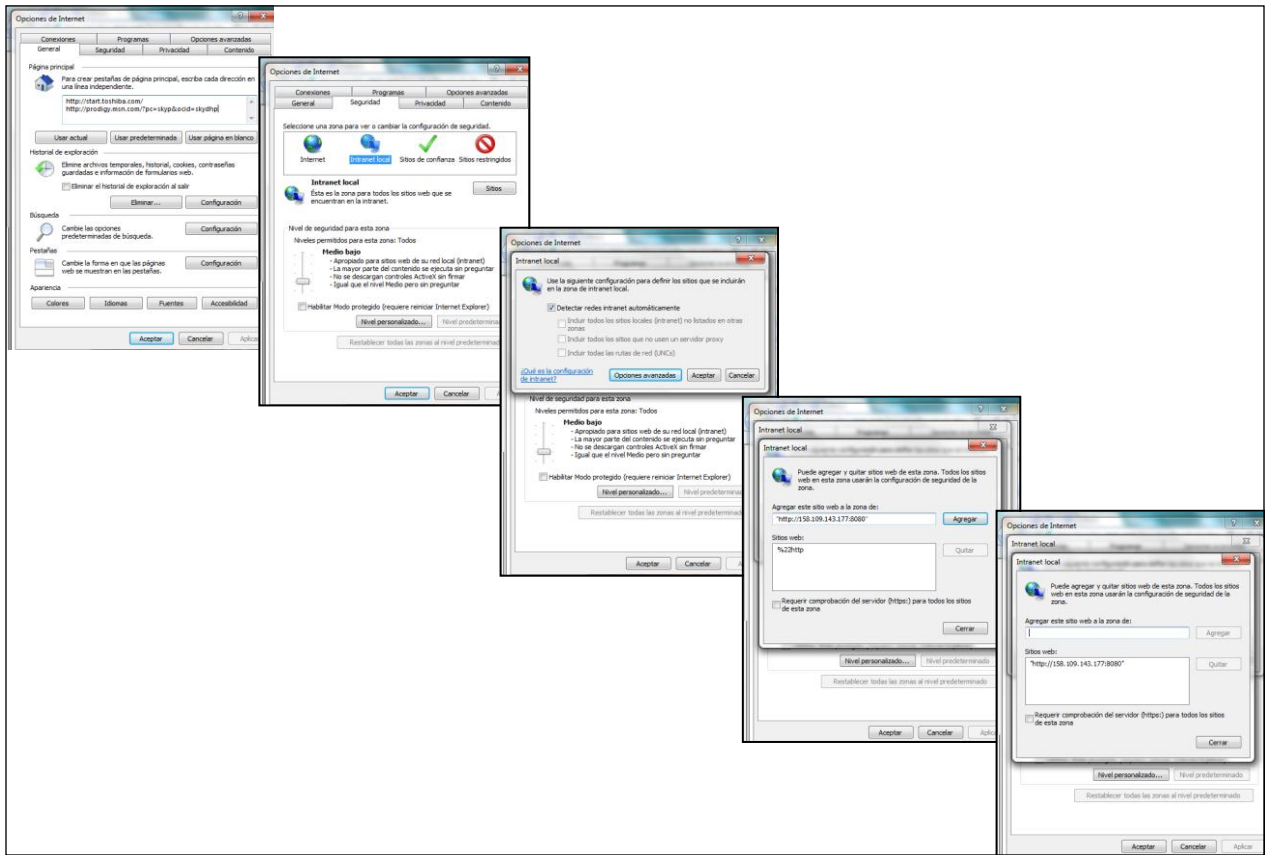
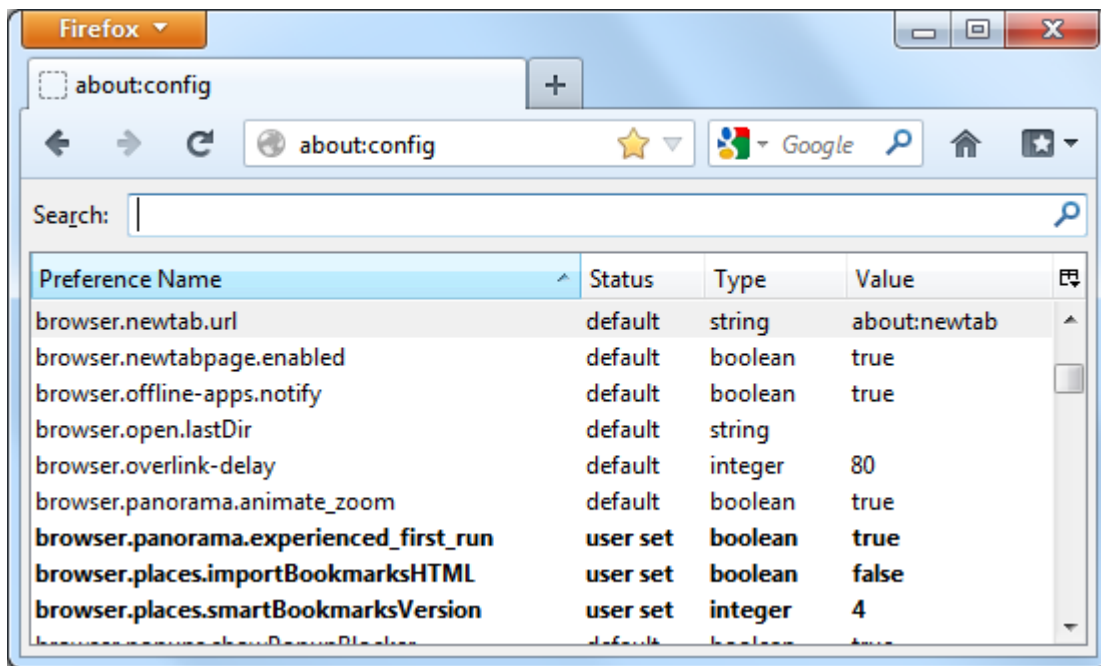


Figura 43. Pasos para la configuración de Internet Explorer

Mozilla Firefox:



Después de colocar en el buscador about:config continúan los siguientes pasos.

Herramientas ---->Opciones---->Avanzado---->General---->Editor de configuración (botón)

Posteriormente al haber realizado los pasos anteriores se tienen las nuevas preferencias las cuales se muestran en el texto siguiente:

```
As alternative, open about:config on Firefox and add the following new entries:  
- name: "capability.policy.localfilelinks.checkloaduri.enabled", string type (cadena in Catalan), value: "allAccess"  
  
- name: "capability.policy.localfilelinks.sites", string type (cadena in Catalan), value: "http://158.109.143.177:8080"  
  
- name: "capability.policy.policynames", string type (cadena in Catalan), value: "localfilelinks"  
  
- name: "network.protocol-handler.expose.file", boolean type, value: false  
  
- name: "network.protocol-handler.external.file", boolean type, value: true
```

Estas preferencias son añadidas de la siguiente manera, primero se realiza click derecho en cualquier lugar de la lista de preferencias. En el menú contextual, se selecciona *Nuevo* y luego se selecciona el tipo de preferencia que se está agregando.

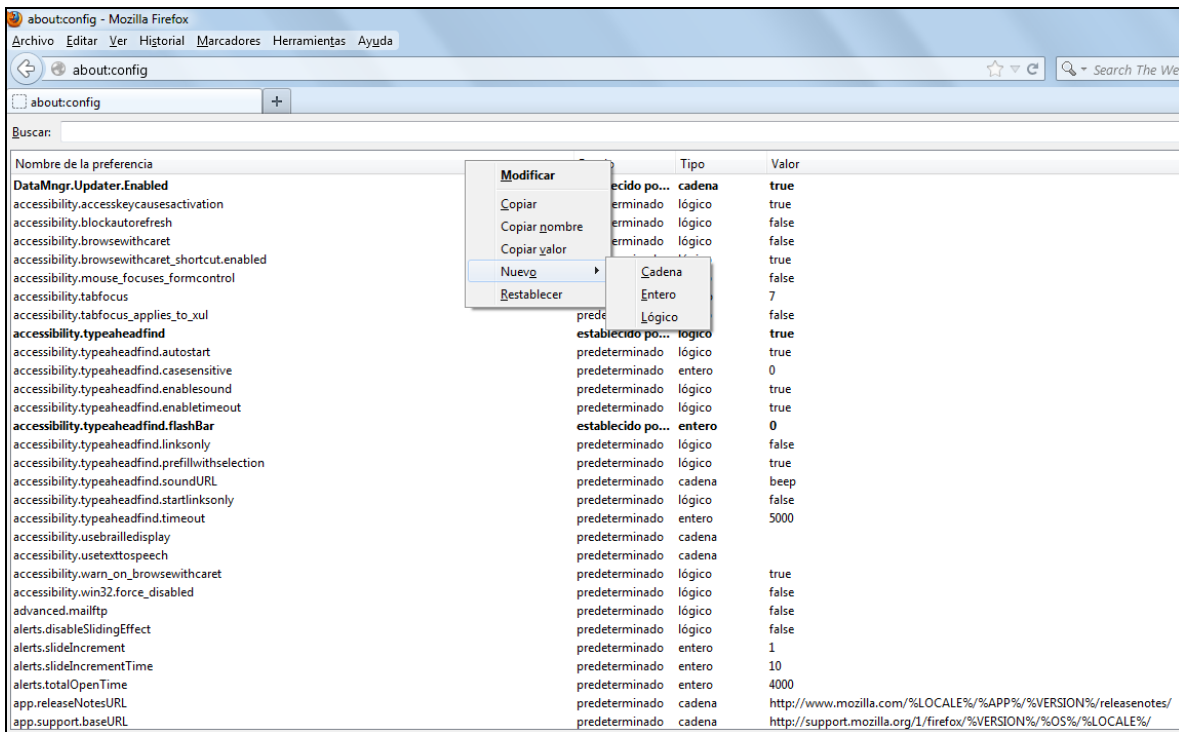
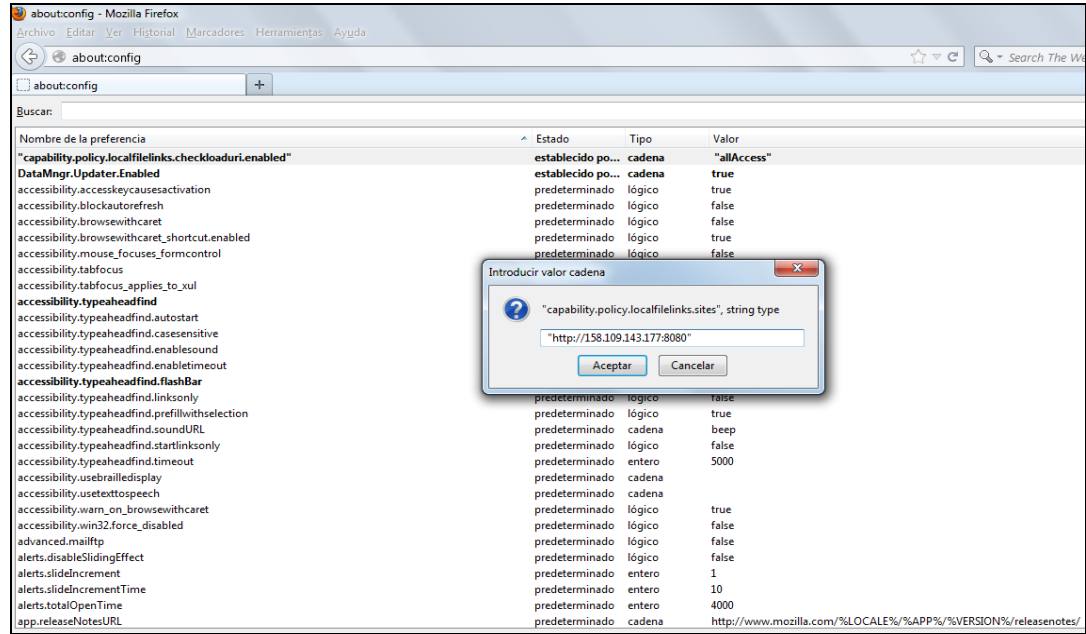
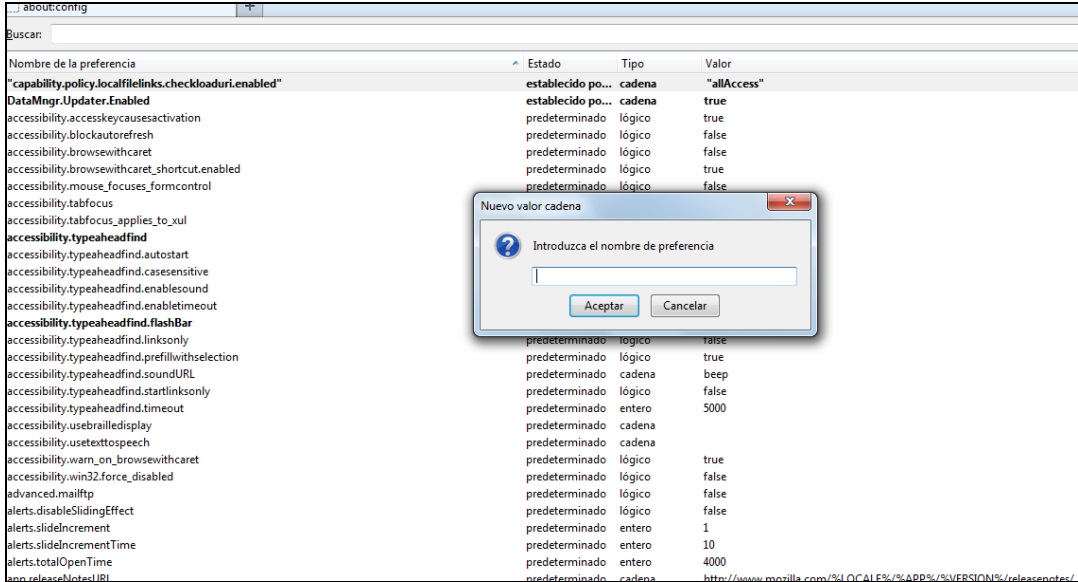


Figura 44. Pasos para la configuración de Mozilla Firefox



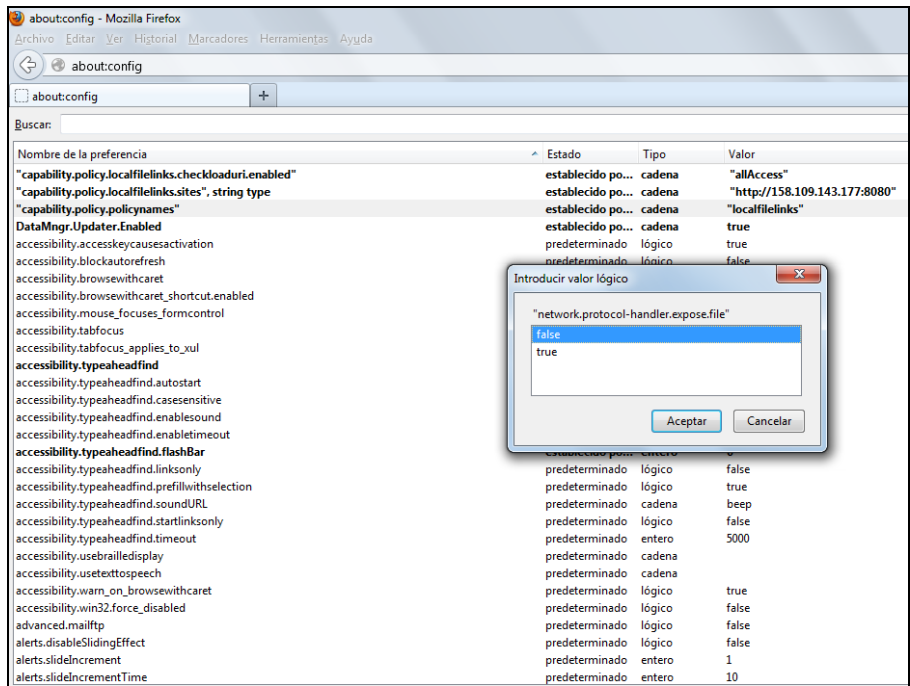
Añadimos cada una de las preferencias que se señalaron anteriormente, dependiendo se escoge si es cadena, entero o booleano.

1. Introducir el nombre de la preferencia.
2. Valor de la preferencia
3. En el caso de ser una preferencia lógica se escoge *True* o *False*

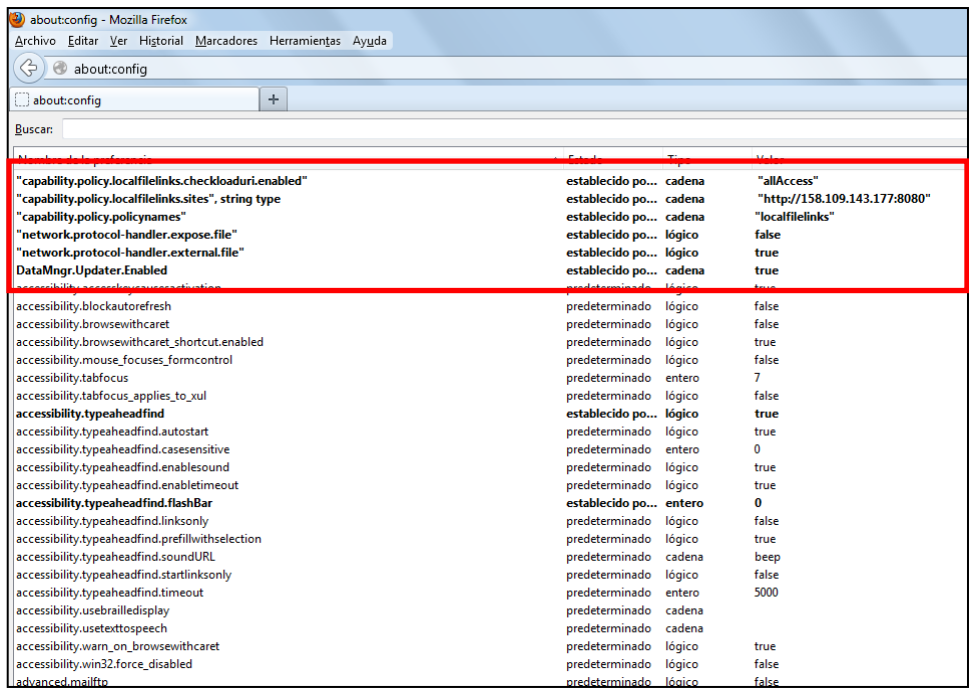




En el caso donde sea una preferencia lógica se realiza los siguientes pasos. Se hace lo mismo que el anterior con la diferencia que se escoge true o false.

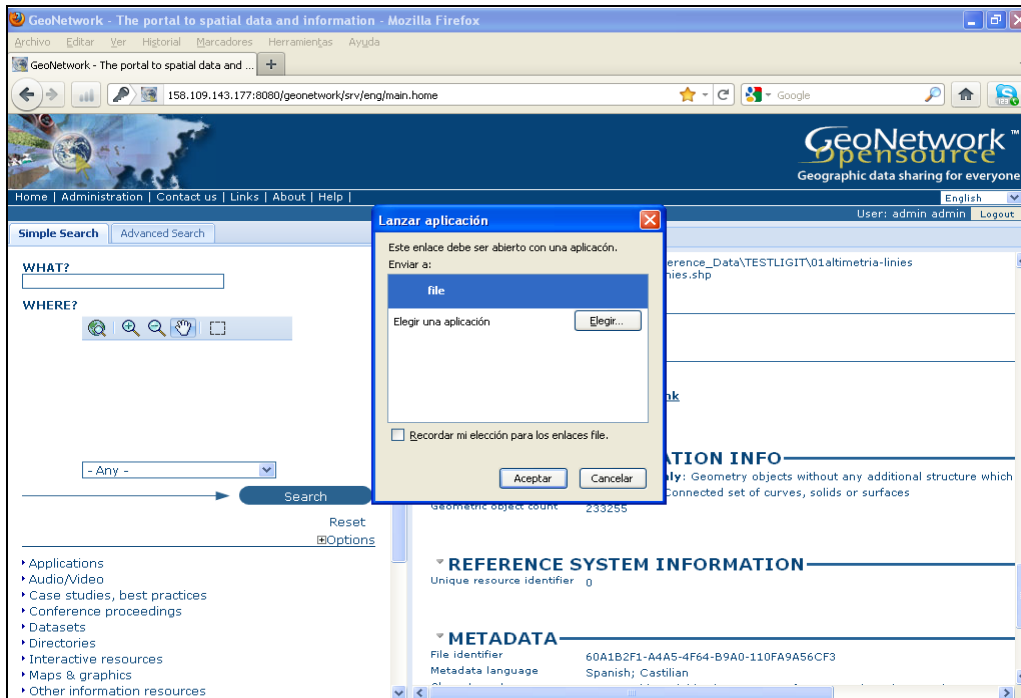


Finalmente las preferencias añadidas quedan de la siguiente manera.



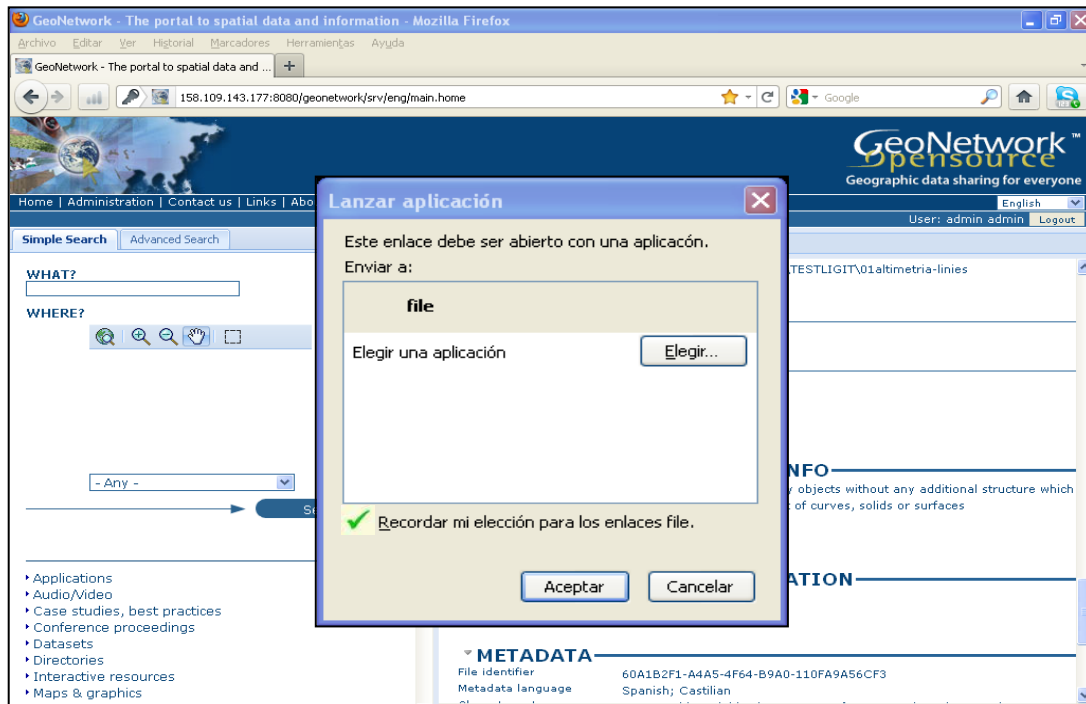
Después de haber realizado las correspondientes configuraciones en los dos exploradores, se procede a acceder a la ruta de los datos directamente desde Geonetwork

### Acceder desde Firefox:

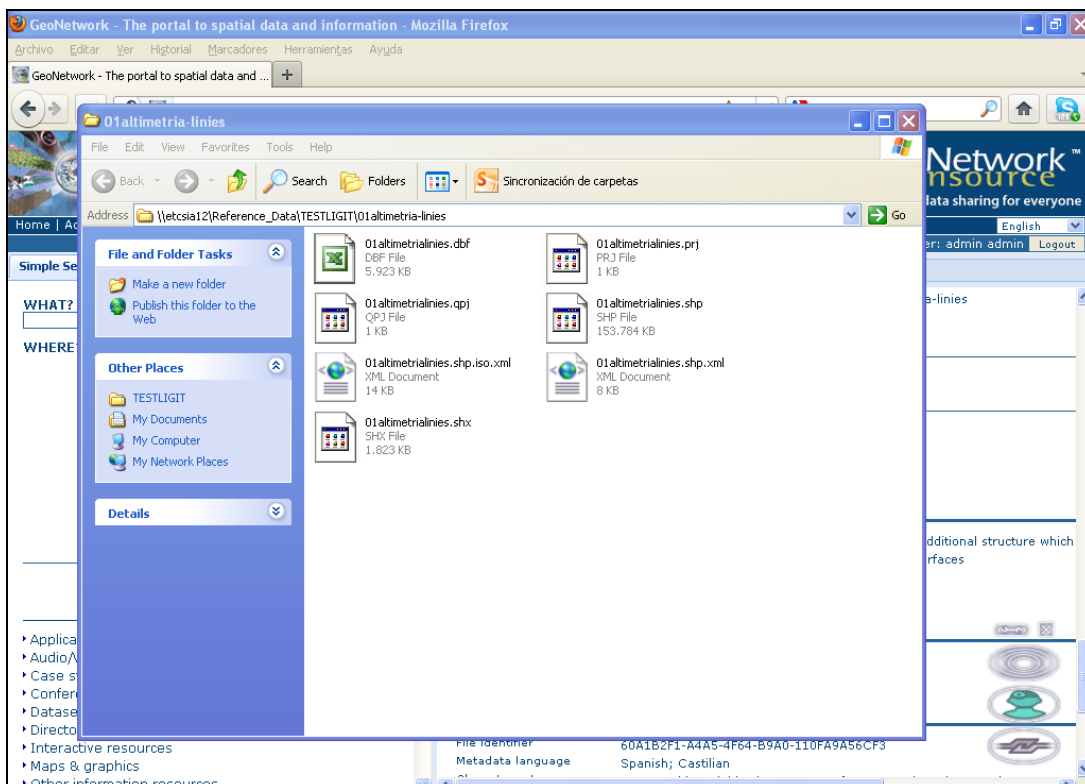


**Figura 45.** Acceso directo a los datos desde Mozilla Firefox

Para que no vuelva a aparecer el recuadro con el mensaje, señalamos donde aparece recordad mi selección para los enlaces file. De esta manera cada vez que accedamos directamente a los datos desde la ruta que fue configurada en Geonetwork, no aparecerá este recuadro y nos llevará directamente a los datos.



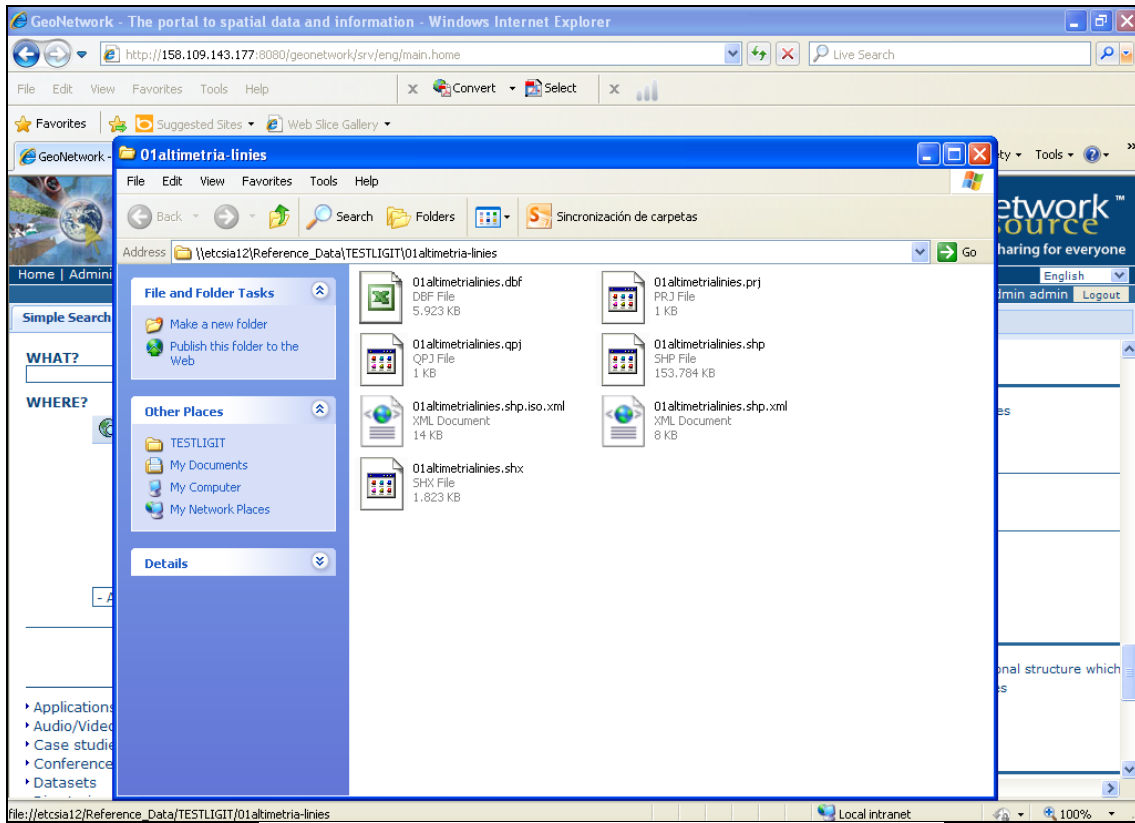
Resultado final, obtención de los datos a través de la configuración de una ruta en el catálogo de Geonetwork.



**Figura 46.** Visualización directa de los datos desde Mozilla Firefox

## Acceder desde Internet Explorer:

Desde Internet Explorer una vez configurado este, se ejecutando dándole click directamente desde la ruta en catálogo de Geonetwork.



**Figura 47.** Acceso directo a los datos desde Internet Explorer

## **7 CONCLUSIONES**

La fase de documentación y análisis del sistema ha permitido conocer los diferentes programas y conceptos que fueron de uso durante el proyecto.

La realización del proyecto puso en práctica muchos de los conocimientos adquiridos durante el periodo teórico del Master de Tecnologías de Información Geográfica, y también permitió adentrarnos en el mundo de la programación conociendo nuevos lenguajes como lo fue Python.

La fase donde se llevó a cabo la publicación de metadatos y el acceso directo a la ruta de los datos directamente desde el catálogo de Geonetwork, fue la más larga y complicada durante la realización del proyecto.

El proyecto se realizó de manera satisfactoria ya que se cumplió con todos los objetivos y fases propuestas, y se realizó durante el tiempo reglamentario. La configuración de Geonetwork permitió crear un acceso directo de la ruta a los datos esto supondrá una manera más rápida y cómoda para que los usuarios y los miembros integrantes del Centro Temático (ETC-SIA) puedan acceder a los datos.

## **8 RECOMENDACIONES**

Se recomienda que a la hora de llenar los campos de los metadatos estos se hagan cumpliendo con los principales campos que exige ArcGis, para su posterior validación y publicación en el catálogo de Geonetwork.

Se recomienda realizar una programación en Geonetwork, de una herramienta de exportación en Geonetwork que sea capaz de crear un listado en formato Excel, incluyendo un breve resumen de los datos y de la ruta.

Se recomienda realizar posteriores estudios con el fin de mejorar y ampliar el funcionamiento de la herramienta masiva de importación diseñado para la edición y publicación de metadatos.

Igualmente se recomienda implementar el uso de lenguaje de Python para futuras programaciones, ya que es un lenguaje que contiene una sintaxis clara, simple y sencilla;

## 9 BIBLIOGRAFÍA

**Amaro A**, 2007 Validación de ficheros XML de metadatos de información geográfica. Norma ISO19139. Área de Teledetección. Departamento de Observación de la Tierra. INTA.

**ArcGis Resource** <http://www.help.arcgis.com>, <http://www.resources.arcgis.com> (En línea).

**Ballari D. et al**, 2008. Normas sobre metadatos (ISO19115, ISO19115-2, ISO19139, ISO 15836). (Draft) Mapping. vol. 123, p. 48-57. ISSN 1-131-9-100.

**Berlanga, A; et al**, Consideraciones para Reforzar el Valor de los Metadatos en los Objetos de Aprendizaje. Departamento de Informática y Automática, Universidad de Salamanca, 37008 Salamanca, España.

**Brodeur et al**, 2000 ISO and OGC e Open Geospatia Consentium.

**Caplan Priscilla**, 1995. You call it corn, we call it syntax-independent metadata for documentlike objects. The Public Access Computer Systems Review, v. 4, n. 6. Consultado diciembre 12 de 2008, en <http://epress.lib.uh.edu/pr/v6/n4/capl6n4.html>.

**Carbajal-Salcedo J**, 2007. Metadatos Institucional. Instituto Geológico Minero y Metalúrgico (INGEMMET).

**DCE**, 1998. Elementos del estandar de metadatos dublin core.

**Downey A, et al**. Libro (PDF) Aprenda a pensar como un programador con Python.

**Duque-Gonzalez R**. Libro (PDF) Python para todos. <http://mundogeek.net/tutorial-python/>

**European Topic Centre (ETC)**, 2010. Spatial Information and Analysis.

**Geonetwork Opensource**. <http://geonetwork-opensource.org/>

**Manual de GeoNetwork Opensource**, 2007-2009.

## **Normalización y Estandarización (En Línea)**

<http://www.iso.org/iso/home.html>

<http://www.isotc211.org/>

<http://www.fgdc.gov/>

<http://www.fgdc.gov/metadata/csdgm>

<http://www.opengeospatial.org/>

<http://www.w3.org/>

<http://www.w3c.es>

<http://www.w3.org/TR/WD/-xml-lang#sec1.1>

**Paulus-Vasquez C**, METADATOS: Introducción e historia.

**Pascual-Rodríguez. A**, *et al.* La Norma ISO19115 de Metadatos: Características y Aplicabilidad.

**Trilles-Oliver, S.** Un geoprocesamiento para la publicación de metadatos geográficos. Institute of New Imaging Technologies Universitat Jaume I.



## 10 ANEXOS

### Script

- **Recorrido de todas las carpetas donde existen metadatos**

```
def process Folder(folder, recurseDirectories=True):  
    logging.info (" * Processing folder: " + folder + "...")  
    Print " * Processing folder: " + folder + "..."  
    folderbak = env.workspace  
    env.workspace = folder  
    doExportFC(folder)  
    doExportDataset(folder)  
    if (recurseDirectories):  
        wpList = arcpy.ListWorkspaces("", "ALL")  
        for wp in wpList:  
            processFolder(wp, recurseDirectories)  
    env.workspace = folderbak
```

- **Recorre los diferentes ficheros y exporta los metadatos a formato XML ISO 19139**

#### **def doExportFC(folder)**

```
direct = arcpy.GetInstallInfo("desktop")["InstallDir"]  
translator = direct + "Metadata/Translator/ArcGIS2ISO19139.xml"  
folderbak = env.workspace  
env.workspace = folder
```

```

fcList = arcpy.ListFeatureClasses()

for fc in fcList:

    desc = arcpy.Describe(fc)

    print folderbak+ "\\ " + fc + ".iso.xml"

    if os.path.exists(folderbak+ "/" + fc + ".iso.xml"):

        os.remove(folderbak + "/" + fc + ".iso.xml")

    #logging.info(" ** Processing fc: " + fc + "...")

    try:

        arcpy.ExportMetadata_conversion (fc, translator, folderbak+"/"+ fc + ".iso.xml")

        insertDataPath(folderbak+ "\\ " + fc, folderbak+ "\\ " + fc + ".iso.xml")

    except:

        exc,value,tcb = sys.exc_info()

        print str(exc) + "\n" + str(value)

        traceback.print_tb(tcb, 10)

    print "FC"

    print "fc " + fc

    print "FIN-----"

folderbak = env.workspace

```

**def doExportDataset(folder):**

```

dir = arcpy.GetInstallInfo("desktop")["InstallDir"]

translator = dir + "Metadata/Translator/ArcGIS2ISO19139.xml"

```

```

folderbak = env.workspace

env.workspace = folder

print "folder: " + folder

desc = arcpy.Describe(folder)

datasets = arcpy.ListDatasets("")

for dataset in datasets:

    logging.info("** Processing dataset: " + dataset + "...")

    print "** Processing dataset: " + dataset + "..."

    desc = arcpy.Describe(folder+"/"+dataset)

    if desc.dataElementType == "DEFeatureDataset":

        arcpy.env.workspace = folder + "\\\" + dataset

        print "dataset: " + dataset

        arr = folder.split("\\")

        num = len(arr)

        nombregdb = arr[num - 1]

        work=""

        for part in arr:

            work = work + part + "\\\"

            work = work.replace("\\\" + nombregdb + "\\\", ")

        featureClasses = arcpy.ListFeatureClasses()

```

```

for fc in featureClasses:

    arcpy.env.workspace = work

    if os.path.exists(work + "/" + fc + ".iso.xml"):

        os.remove(work + "/" + fc + ".iso.xml")

    try:

        arcpy.ExportMetadata_conversion (nombregdb + "/" + dataset + "/" + fc,
translator,

nombregdb + "_" + dataset + "_" + fc + ".iso.xml")

        if os.path.exists(work + "/" + nombregdb + "_" + dataset + "_" + fc +
".iso.xml"):

            insertDataPath(nombregdb + "/" + dataset, work + "/" + nombregdb + "_" +
dataset + "_" + fc + ".iso.xml")

        else:

            print "Metadata could not be exported: " + work + "/" + nombregdb + "_" +
dataset + "_" + fc + ".iso.xml"

    except:

        exc,value,tcb = sys.exc_info()

        print str(exc) + "\n" + str(value)

        traceback.print_tb(tcb, 10)

    print "FC gdb"

    print "fc " + fc

    print "FIN-----"

env.workspace = folderbak

```

➤ **Configuración de acceso a la ruta a los datos directamente desde el catálogo de Geonetwork.**

```
def insertDataPath(datapath, isoxml):

    tree = ET.parse(isoxml)

    root = tree.getroot()

    ET._namespace_map["http://www.isotc211.org/2005/gmd"] = "gmd"

    ET._namespace_map["http://www.isotc211.org/2005/gco"] = "gco"

    ET._namespace_map["http://www.w3.org/2001/XMLSchema-instance"] = "xsi"

    ET._namespace_map["http://www.isotc211.org/2005/srv"] = "srv"

    ET._namespace_map["http://www.opengis.net/gml"] = "gml"

    ET._namespace_map["http://www.isotc211.org/2005/gts"] = "gts"

    ET._namespace_map["http://www.fao.org/geonetwork"] = "geonet"

    root.set("xmlns:gco", "http://www.isotc211.org/2005/gco")

    if len(datapath)>3:

        urlDataPath = BaseDataPathURL + datapath[2:].replace("\\", "/")

        winDataPath = BaseDataPathWin + datapath[2:]

    distInfo = root.find('{http://www.isotc211.org/2005/gmd}distributionInfo')

    if not distInfo:

        distInfo = ET.SubElement(root,
        '{http://www.isotc211.org/2005/gmd}distributionInfo')

        mdDistInfo = distInfo.find('{http://www.isotc211.org/2005/gmd}MD_Distribution')

    if not mdDistInfo:
```

```

mdDistInfo =
ET.SubElement(distInfo, '{http://www.isotc211.org/2005/gmd}MD_Distribution')

#distInfo = ET.SubElement(root,
'{http://www.isotc211.org/2005/gmd}distributionInfo')

#mdDistInfo = ET.SubElement(distInfo,
'{http://www.isotc211.org/2005/gmd}MD_Distribution')

#distFormat = ET.SubElement(mdDistInfo,
'{http://www.isotc211.org/2005/gmd}distributionFormat')

distFormat = ET.Element('{http://www.isotc211.org/2005/gmd}distributionFormat')

mdDistInfo.insert(0, distFormat)

mdDistFormat = ET.SubElement
(distFormat, '{http://www.isotc211.org/2005/gmd}MD_Format')

name = ET.SubElement (mdDistFormat, '{http://www.isotc211.org/2005/gmd}name')

charstr = ET.SubElement(name, '{http://www.isotc211.org/2005/gco}CharacterString')

charstr.text = winDataPath

versgco = ET.SubElement(mdDistFormat, '{http://www.isotc211.org/2005/gmd}version
gco:nilReason="missing")

specgco = ET.SubElement(mdDistFormat,
'{http://www.isotc211.org/2005/gmd}specification gco:nilReason="missing")

transOption = ET.SubElement (mdDistInfo,
'{http://www.isotc211.org/2005/gmd}transferOptions')

mdDigiTransOption = ET.SubElement
(transOption, '{http://www.isotc211.org/2005/gmd}MD_DigitalTransferOptions')

onLin = ET.SubElement (mdDigiTransOption,
'{http://www.isotc211.org/2005/gmd}onLine')

ciOnlinReso = ET.SubElement (onLin,
'{http://www.isotc211.org/2005/gmd}CI_OnlineResource')

link = ET.SubElement (ciOnlinReso, '{http://www.isotc211.org/2005/gmd}linkage')

Url = ET.SubElement (link, '{http://www.isotc211.org/2005/gmd}URL')

```

```

Url.text = urlDataPath

proto = ET.SubElement (ciOnlinReso, '{http://www.isotc211.org/2005/gmd}protocol')

chastr = ET.SubElement (proto, 'gco:CharacterString')

chastr.text = "FILE:GEO"

name = ET.SubElement (ciOnlinReso, '{http://www.isotc211.org/2005/gmd}name')

chastr = ET.SubElement (name, 'gco:CharacterString')

chastr.text = "File link"

descrgco = ET.SubElement(ciOnlinReso,
'{http://www.isotc211.org/2005/gmd}description gco:nilReason="missing"')

onLine = ET.SubElement(mdDigiTransOption,
'{http://www.isotc211.org/2005/gmd}onLine')

ciOnlinResour = ET.SubElement
(onLine, '{http://www.isotc211.org/2005/gmd}CI_OnlineResource')

linka = ET.SubElement (ciOnlinResour, '{http://www.isotc211.org/2005/gmd}linkage')

Url = ET.SubElement (linka, '{http://www.isotc211.org/2005/gmd}URL')

Url.text = os.path.dirname(urlDataPath) # la url del directorio

proto = ET.SubElement (ciOnlinResour, '{http://www.isotc211.org/2005/gmd}protocol')

charstr = ET.SubElement(proto, '{http://www.isotc211.org/2005/gco}CharacterString')

charstr.text = "WWW:LINK-1.0-http--link"

name = ET.SubElement (ciOnlinResour, '{http://www.isotc211.org/2005/gmd}name')

charstr = ET.SubElement(name, '{http://www.isotc211.org/2005/gco}CharacterString')

charstr.text = "Directory link"

descript = ET.SubElement(ciOnlinResour,
'{http://www.isotc211.org/2005/gmd}description gco:nilReason="missing"')

tree.write(isoxml)

```

- **Configuración de observación del tiempo que durara la ejecución total del script. Carpeta raíz donde encontraban los metadatos es decir el espacio de trabajo.**

```
import sys, time, logging

import traceback

def main():

    try:

        print "Starting time:"+ str(time.localtime(time.time()))

        inputFolder = "Z:\\"

        processFolder(inputFolder)

    except:

        exc,value,tcb = sys.exc_info()

        logging.error(str(exc) + "\n" + str(value))

    raise

    print "Finish time:"+ str(time.localtime(time.time()))

if __name__ == "__main__":

    main()
```