Master Thesis | Tesis de Maestría

submitted within the UNIGIS MSc programme
Presentada al Programa UNIGIS MSc

at/en

Interfaculty Department of Geoinformatics - Z_GIS
Departamento de Geomática – Z_GIS
University of Salzburg | Universidad de Salzburgo

DESCARGA AUTOMÁTICA DE INFORMACIÓN ESPACIAL MEDIANTE UNA RUTINA PYTHON PARA LA CIUDAD DE CALI, COLOMBIA

AUTOMATIC DOWNLOAD OF SPATIAL INFORMATION THROUGH A PYTHON ROUTINE FOR CALI COLOMBIA

By/por

Jesús Antonio Pantoja Cueltán
01223196

A thesis submitted in partial fulfilment of the requirements of
the degree of
Master of Science (Geographical Information Science & Systems) – MSc (GIS)

Advisor | Supervisor:

Anton Eitzinger. PhD

Bogotá D.C.- Colombia, agosto de 2020

COMPROMISO DE CIENCIA

Por medio del presente documento, incluyendo mi firma personal certifico y aseguro que mi tesis es completamente el resultado de mi propio trabajo. He citado todas las fuentes que he usado en mi tesis y en todos los casos he indicado su origen.

Dedicatoria

Antonio Pantoja C

A la comunidad UNIGIS, tanto a sus estudiantes como a profesores y cuerpo administrativo en Colombia, América latina y Salzburgo.

Igualmente a las instituciones y miembros de la Comunidad de Práctica Andes (COPANDES).

Agradecimiento

Agradezco a Patricia Pantoja, Lorena Gómez, Mary Lugo y Emmanuel Zapata por su apoyo pedagógico y personal.

También a Laure Collet, Anton Eitzinger y Karl Atzmanstorfer por sus aportes académicos y brindarme la oportunidad de pertenecer a la comunidad UNIGIS

¡Gracias!

Resumen

El crecimiento acelerado de la información genera gran cantidad de datos que requieren ser procesados a gran escala y, en lo posible, de manera automática. Sin embargo, actualmente no se tiene un nivel óptimo de integración entre las plataformas que ofrecen la información y las herramientas que el usuario puede utilizar para acceder a ella—obstruyendo la adecuada recopilación y el procesamiento de esos datos—.

El presente estudio evaluó la accesibilidad a la información geográfica del Web Feature Service (WFS) ofrecido por la Infraestructura de Datos Espaciales de Cali (IDESC), Colombia. Dicho proceso se hizo mediante la aplicación de una rutina en el lenguaje de programación Python para descargar información de manera automática. Esta rutina se implementó con dos propósitos: (i) determinar los nombres de las capas y (ii) descargar los archivos disponibles.

El script creado permitió leer el documento resultante de la solicitud GetCapabilities —del WFS— para conocer el contenido y los nombres de los archivos de la IDESC. La metodología resultó viable y se aplicó con éxito en la IDESC. Sin embargo, al intentar utilizarla en otras infraestructuras de datos espaciales (IDE) se presentaron algunos problemas con los nombres de las capas. Esta situación se debe a la falta de algún estándar específico en la nomenclatura de los archivos alojados en los repositorios WFS.

Por lo tanto, al intentar generar una rutina de acceso automático al WFS de una IDE cualquiera (diferente de la presentada en este estudio), es necesario tener en cuenta excepciones en el código para no obtener errores. En consecuencia, es clara la necesidad de implementar estándares que consideren la automatización del acceso y descarga de grandes volúmenes de información geográfica desde las IDE.

Palabras Clave: IDE, Python, GetCapabilities, Descarga, Web Feature Service (WFS).

.

Abstract

The accelerated growth of information generates a vast amount of data that needs to be processed on a large scale and, if possible, automatically. However, there is not an optimal level of integration between the platforms that offer the information and the tools that the user can use to access it, obstructing the proper collection and processing of this data.

This study assessed the accessibility to geographic information of the Web Feature Services (WFS) offered by the Infrastructure of Spatial Data of Cali (IDESC, by its acronym in Spanish), Colombia. The assessment was carried out through the application of a routine in Python programming language to download data automatically. Such a routine was implemented based on two purposes: (i) determining the names of the layers and (ii) downloading the available files. All automatically.

The developed script allowed reading the resulting document from the GetCapabilities request—from the WFS—to know the content and file names of the IDESC. This methodology proved to be feasible and was successfully applied to IDESC, however, when trying to use it in other spatial data infrastructure (SDI), there were issues related to the layers naming. This situation is due to the lack of any specific standard for the naming of files hosted in WFS repositories.

Therefore, when trying to generate an automatic access routine to the WFS of any IDE (different from the one presented in this study), it is necessary to have into account exceptions in the code in order not to obtain errors. Thus, it is clear there is need for implementing standards that consider automation in the access and download of large volumes of geographic information.

Key words: SDI, Python, GetCapabilities, Downloading, Web Feature Service (WFS).

Índice general

Glosario

API: *Application Programming Interface.*

CEPAL: Comisión Económica para América Latina y el Caribe.

CERN: European Organization for Nuclear Research, (es español: Organización Europea para la Investigación Nuclear.

CP IDEA: Comité Permanente para la Infraestructura de Datos de las Américas.

CQL: *Common Query Lang.* El lenguaje de filtros para WFS, que se define con un XML derivado de CQL.

CRISP-DM: Cross-*Industry Standard Process for Data Mining*, (en español: procesos estándares para el uso de minería de datos).

DXF: Drawing *Exchange Format*, (en español: formato de intercambio de dibujo).

ESRI: Environmental Systems Research Institute

FGDC: *Federal Geographic Data Committee*, (en español: Comité Federal de Datos Geográficos).

FSF: *Free Software Foundation (en español: fundación del software libre).*

GC*: Grid Coverages* (en español: cobertura de cuadrículas).

GML: *Geography Markup Language* (GML) (en español: lenguaje demarcado de geografía).

GNU GPL: GNU General *Public License* (en español: Licencia Pública General de GNU).

GSDI: Global Spatial Data Infrastructure Association (en español: Asociación Global de Infraestructuras de Datos Espaciales).

HTTP: *Hypertext Transfer Protocol* (en español: Protocolo de transferencia de hipertexto).

IA: Inteligencia Artificial

IBM: International Business Machines Corporation (es una multinacional estadounidense de tecnología y consultoría).

IDE: Infraestructura de Datos Espaciales.

ICDE: Infraestructura de Datos Espaciales de Colombia.

IDECA: Infraestructura de Datos Espaciales de Bogotá.

IDESC: Infraestructura de Datos Espaciales de la ciudad de Cali, Colombia.

IDS: *Informix Dynamic* Server (Informix es una familia de productos de IBM).

IDLE: *Integrated DeveLopment Environment (*en español: entorno de desarrollo integrado).

IG: Información Geográfica

OEA: Organización de los Estados Americanos

OGC: *Open Geospatial Consortium* (en español: Consorcio Geoespacial Abierto).

POO: Programación orientada a objetos

RedIRIS: Red Académica y de Investigación Española

SGC: Servicio Geológico Colombiano

SHP: *Shapefile* (en español: archivo de forma).

SIG: Sistema de Información Geográfica (del original en inglés: GIS - *Geographic Information System*).

SLD: *Styled Layer Descriptor* (en español: descriptor de capa con estilo).

SNIT: Sistema Nacional de Información Territorial de Costa Rica

SOM: Mapa Auto-organizado

URL: *Uniform Resource Locator* (en español: localizador uniforme de recursos).

XML: *Extensible Markup Language* (en español; lenguaje de marcas Extensible).

WCS: *Web Coverage Service (*en español: servicio web de coberturas).

WFS: *Web Feature Service* (en español: servicio de entidades vectoriales).

WFS-T: *Transactional Web Features Service* (en español: servicio de entidades vectoriales transaccional).

WMC: *Web Map Context* (en español: servicio web de contexto de mapas).

WMS: Web Map Service (en español: servicio web de mapas).

WKT: *Well Know Text* (en español: representación de texto conocido).

Índice de Figuras

Índice de Tablas

1. INTRODUCCIÓN

1.1. ANTECEDENTES DEL PROBLEMA

Según Montes (2008), el desarrollo de los Sistemas de Información Geográfica (SIG) empieza a evidenciarse en mayor magnitud en los años 70, donde se genera una serie desarrollos paralelos en las tecnologías relacionadas con los SIG, y de igual manera se da una serie de cambios en la manera de acceder a la información georreferenciada, la cual estuvo cada vez más disponible y al alcance de las universidades, organismos de investigación y el sector privado.

Entonces, la sociedad actual lleva más de medio siglo en un proceso de digitalización. Sin embargo, no se tiene un nivel óptimo que permita integrar diferentes fuentes de datos para generar información a partir de la ya existente. Esto puede deberse, entre otras razones, a las dificultades para aplicar políticas de accesibilidad a los datos digitales y la inadecuada estructuración entre las autoridades y los organismos correspondientes. Por otro lado, se ha creado muchos estándares industriales por parte de algunas empresas que buscan proteger su sistema de manejo de la información, como por ejemplo SHP (*Shapefile* de ESRI), DXF (*Drawing Exchange Format* de AUTOCAD), entre otros.

Desde hace poco menos de una década está disponible una cantidad considerable de datos geográficos debido a la migración de los datos análogos a digitales, que son cada vez más accesibles gracias a las políticas de datos abiertos que los productores de dicha información están implementando. Sin embargo, usualmente no es posible descargar dicha información de manera rápida y completa, porque no hay derechos para la descarga libre o bien porque se desconocen los canales adecuados para acceder a ella.

Las Infraestructuras de Datos Espaciales (IDE), pretenden ser una solución a la necesidad de mejorar el acceso a lo que se define como datos geográficos de referencia o datos fundamentales. Para ello, organizaciones como el *Open Geospatial Consortium* (OGC), han invertido mucho esfuerzo en definir estándares para la estructuración e intercambio de datos digitales, entre ellos, el estándar WFS

(*Web Feature Service*) para que se permita de manera libre y organizada consultar y descargar coberturas geográficas a través de las IDE.

Aunque países como los Estados Unidos, el Reino Unido, Australia y España son pioneros en el desarrollo de las IDE, Colombia inició en 1996 con el proceso de consolidación de la Infraestructura Colombiana de Datos Espaciales (ICDE), por lo cual no es un tema reciente. Aunque ciertamente con el crecimiento de la disponibilidad de la información se hace necesario desarrollar e implementar nuevas herramientas informáticas para mejorar la accesibilidad a la información espacial y así facilitar la descarga de los datos de manera automática y total directamente desde un servicio como el WFS.

En el año 2009 se creó la Infraestructura de Datos Espaciales de Cali (IDESC), con la finalidad de permitir una eficiente gestión de la información geográfica (IG) del municipio de Santiago de Cali (Colombia), definiendo como prioridad “armonizar los procesos de captura, análisis, acceso, uso y distribución de la IG que ejecutan los organismos de la administración municipal” (IDESC, 2009, ¶1).

Las posibilidades para descargar IG desde el servicio WFS de la IDESC se limita a las herramientas tradicionales mediante el uso de programas SIG de escritorio, los cuales no permiten conocer de manera rápida la cantidad total de archivos alojados en el servicio, ni tampoco descargar los datos de manera automatizada; por esta razón se realiza este estudio de caso, diseñando una metodología donde a través de un script o rutina se permita listar y descargar de manera automática la información desde servicios WFS, aplicándolo a la IDE de Cali, Colombia. Esto será útil para los casos en los que se requiera procesar gran cantidad de información de manera local, es decir, en un computador o servidor propio del usuario.

1.2. OBJETIVO GENERAL

Evaluar la accesibilidad de la información del servicio WFS publicada en la IDE, de Cali, Colombia, mediante la aplicación de una metodología para listar y descargar de manera automática dicha información geográfica.

1.3. OBJETIVOS ESPECÍFICOS

- Verificar la disponibilidad de información geográfica en el servicio WFS de la Infraestructura de Datos Espaciales de Cali (IDESC).
- Desarrollar un método para el acceso y descarga automática de la información geográfica del servicio WFS de la IDESC.
- Evaluar la utilidad del método desarrollado, para el acceso y descarga de la información del servicio WFS de la IDESC.

1.4. PREGUNTAS DE INVESTIGACIÓN

La presente investigación se sustenta en el diseño de una rutina o script aplicado a la IDE de la ciudad de Cali, Colombia, que permita descargar las coberturas disponibles en el servicio WFS, con la finalidad de evaluar la accesibilidad de la información disponible bajo ese estándar. Para ello se generan las siguientes interrogantes:

- ***«PI - 1»*** ¿Cuál es la disponibilidad de la información geográfica en el servicio WFS de la IDESC?
- ***«PI- 2»*** ¿Se puede aplicar un método para acceder a la información geográfica y descargarla de manera automática desde el servicio WFS de las IDESC?
- ***«PI - 3»*** ¿Cuáles serían las ventajas de aplicar un método de consulta y descarga de información de manera automática desde el WFS de la IDESC?

1.5. HIPÓTESIS

Mediante el uso de una metodología y la aplicación de una rutina, basada en el documento resultante de la petición GetCapabilities en el servicio WFS, de la IDESC; se puede generar el URL de descarga de cada capa, y así descargar los datos disponibles en dicho servicio con un proceso automatizado.

1.6. JUSTIFICACIÓN

El aumento de la producción de datos espaciales, así como de su demanda, no solo por parte de organismos públicos sino también por empresas privadas e incluso particulares, hace imperativa la creación y adecuación de herramientas que permitan al usuario acceder y hacer uso más eficiente de la información contenida en los geoportales.

Para Bermejo y Conti (s.f., ¶ 3) “La accesibilidad del usuario a la información de forma gratuita, no solo mediante la visualización sino también de su descarga, ha significado un aumento de la demanda que necesita ir a la par de un sistema de organización y comparación de datos.” Por otra parte, según Mascarell (2014, p.10), “Una IDE debe contener también servicios para descubrir qué datos hay disponibles, servicios para acceder a la información geográfica y a la cartografía, y finalmente metadatos que describan los conjuntos de datos y los servicios disponibles”.

La Infraestructura de Datos Espaciales de Cali (IDESC) es una IDE de nivel local que presta sus servicios al municipio de Cali, en el suroccidente colombiano, siendo hasta el momento la única IDE en dicha zona del país. Dada su estratégica ubicación geográfica se constituye en una entidad de alta importancia en los proyectos que demandan este tipo de servicios y de información. Actualmente la IDESC carece de herramientas que permitan la descarga automática de todas las coberturas del servicio WFS en un solo proceso de manear automática. Esto último ayudaría a responder a la necesidad de los usuarios que deseen tener una copia de los datos de manera local, de forma rápida y completa para realizar análisis que involucren algún tipo de procesamiento en su propio equipo de cómputo.

Tomando en cuenta el panorama actual se plantea y justifica el proyecto de investigación, pues no existe una metodología que oriente la construcción de una rutina para aprovechar el resultado de la petición *GetCapabilities* de un servicio WFS para conocer, en primer lugar, un listado de la información disponible y posteriormente descargarla de manera total.

Esto implicará la posibilidad de que los usuarios de las IDE tenga más herramientas para acceder con agilidad a la descarga de la información, en este caso específico, los usuarios de la IDESC.

1.7. ALCANCE

El proyecto de investigación busca desarrollar y aplicar una propuesta metodológica para implementar un script o rutina para la IDE de la ciudad de Cali, Colombia, hecho en Python, basado en el documento resultante de la petición *GetCapabilitie*s, para descargar todas las coberturas de su servicio WFS, teniendo en cuenta el contexto general de la carencia de una iniciativa similar conocida y de uso libre en esta región.

En este caso de estudio se pretende aplicar y evaluar la viabilidad de la aplicación de esta rutina, y documentar el proceso, de tal forma que a futuro otros usuarios puedan considerar su pertinencia para adaptarla y aplicarla en otras IDE a nivel local, regional o nacional, sirviendo así este documento como base metodológica para ese propósito.

Este proyecto de investigación puede brindar herramientas adicionales a los potenciales usuarios de las IDE y a los profesionales SIG que tengan interés en herramientas de programación (como lo es el lenguaje de programación Python), para simplificar el proceso de descarga de coberturas de servicios WFS y facilitar el acceso y uso de la información geoespacial en los procesos de consulta tanto para ciudadanos como para funcionarios.

En resumen, la rutina de Python desarrollada en esta investigación podrá ser aplicada de tal manera que los usuarios de las IDE puedan descargar automáticamente las capas de información disponibles en los servicios con estándares WFS disponible en la IDESC; los cuales pueden ser utilizados posteriormente para algún requerimiento específico en su campo laboral, complementando los datos que se requieran.

2. MARCO TEÓRICO

2.1. INVESTIGACIONES RELACIONADAS

Demšar y Zupan (2013) desarrollaron el programa ***Orange***, que se basa en el lenguaje programación Python, elegido para su proyecto por tener una sintaxis simple.

Pero a pesar de su simplicidad, Python es ampliamente utilizado en la industria; por ejemplo, está detrás de muchas de las tecnologías de *Google*.

Estos investigadores desarrollaron una herramienta de exploración de datos en la que pudieran diseñar sus propios canales de análisis de datos. De manera que hicieron una serie de bibliotecas Python para implementar interfaces gráficas de usuario. Los investigadores eligieron **Qt** para la representación visual y permitir la exploración del modelo resultante. Además el usuario puede seleccionar un nodo en un árbol o regla de clasificación y explorar las instancias de capacitación cubiertos por ellos.

Cabe destacar que también se hubiesen podido usar nomogramas para explicar las predicciones del modelo. Esto también se aplica a métodos no supervisados, como reglas de asociación, escalamiento multidimensional, mapas auto organizados y varios tipos de agrupamiento.

Sin embargo ***Orange*** no se limita a este tipo de métodos de aprendizaje automático, sino que tiene además una rica colección de métodos de visualización que incluye diagramas de caja, histogramas y diagramas de dispersión, junto con las multivariadas, incluyendo coordenadas paralelas, visualización de mosaico, diagrama de tamiz y levantamiento de gráficos. El usuario puede explorar de manera interactiva las visualizaciones y hacer uso de otros *widgets* que envían o reciben datos. ***Orange*** también puede ayudar al usuario a encontrar visualizaciones perspicaces, clasificándolas automáticamente por interés u organizándolas en una red de visualizaciones.

Por otro lado, Lemaître, Noguera y Áridas (2016) desarrollaron una herramienta llamada *imbalanced-learn*, o aprendizaje desequilibrado, que consiste en una caja de herramientas de Python para enfrentar la incomodidad de tener conjuntos de datos desequilibrados en el aprendizaje automático. Para garantizar la calidad del código se proporciona un conjunto de pruebas unitarias que conducen a una cobertura del 99% para la versión 0.2 de la caja de herramientas. Además, la consistencia del código se garantiza por el estándar PEP8 y cada nueva contribución se verifica automáticamente, proporcionando métricas relacionadas con la calidad del código.

Dicho desarrollo se basa en la comunidad, todo se realiza de manera colaborativa. Las herramientas como *git, GitHub y gitter* se utilizan para facilitar la programación de distintos programadores, mediante seguimiento, integración de código y discusiones de ideas.

Se proporciona también una documentación de API consistente usando *sphin*x y *numpydoc*. También se proporciona una guía de instalación adicional y ejemplos que están centralizados en GitHub1. El repositorio se visita no menos de 2000 veces por semana, atrayendo a unos 300 visitantes únicos por semana. La caja de herramientas de aprendizaje desequilibrado proporciona cuatro estrategias diferentes para abordar el problema de conjunto de datos desequilibrado: (i) muestreo insuficiente, (ii) muestreo excesivo, (iii) una combinación de ambos, y (iv) aprendizaje conjunto. En dicho trabajo se presentan los fundamentos de la visión de la caja de herramientas de aprendizaje desequilibrado y API. Hacia futuro se agregarán métodos de selección, generación y reducción, así como también guías de usuario adicionales.

Por otra parte, Calderón, Barbosa y Cabezuelo (2016) realizan un documento sobre técnicas B*ig Data:* análisis de textos a gran escala para la investigación científica**,** exponiendo el proceso de análisis de datos de tipo texto a escalas grandes, y el análisis de contenidos de manera automática, definiendo claramente conceptos como: minería de datos (*data mining*), aprendizaje automatizado (*machine learning*), modelamiento de temas (*topic modeling*) y análisis de sentimientos (*sentiment analysis*). Se expone cuál es la infraestructura necesaria para el análisis

de B*ig Data* a través del despliegue de centros de cómputo distribuido y se valora el uso de las principales herramientas para la obtención de información a través de programas de cómputo comerciales y de paquetes de programación como Python y R.

Un antecedente a resaltar es también el de Castro, Sifuentes, González y Rascón (2014), que realizan un desarrollo con el uso de minería de datos en el manejo de información geográfica. En dicho trabajo se presenta una metodología para el manejo de datos sobre información geográfica. La metodología se aplicó en la identificación de materiales de construcción de calles y avenidas de un área geográfica específica de Ciudad Juárez, México. Los datos necesarios para el desarrollo del modelo fueron obtenidos del Sistema de Información Geográfica de la Universidad Autónoma de Ciudad Juárez, los cuales incluyeron la intensidad y las coordenadas X, Y, Z de unos 878 mil registros de la ciudad. El desarrollo se realizó en varias etapas. La primera corresponde al pre-procesamiento y limpieza de los datos; la segunda es la formulación del modelo e incorporación de los datos a un gestor de base de datos; la tercera es el análisis mismo con técnicas de minería de datos utilizando el *software Waika to Environment Knowledge Analysis (*WEKA). Para la predicción se utilizaron los algoritmos k-*means y Make Density Based Clusters.*

2.2. INFRAESTRUCTURA DE DATOS ESPACIALES (IDE)

Según Iniesto et al. (2014), el concepto de Infraestructura de Datos Espaciales se empezó a conocer en los años 90, como un tema relacionado con los sistemas abiertos. Y más precisamente en 1994, siendo presidente de los Estados Unidos Bill Clinton, quien firmó la declaración para crear el Comité Federal de Datos Geográficos (*Federal Geographic Data Committee* – FGDC).

Las IDE surgen por la necesidad de lograr una correcta interoperabilidad de los datos y poder compartir información, especialmente entre los organismos con facultad de decisión, donde, incluso hoy en día todavía existen dificultades para lograrlo.

Según la IDEE (2017, ¶ 1) “Una Infraestructura de Datos Espaciales es un sistema informático integrado por un conjunto de recursos (catálogos, servidores, programas, aplicaciones, páginas web…), que permite el acceso y la gestión de datos y servicios geoespaciales (descritos a través de sus metadatos), disponibles en servidores remotos, que cumple una serie normas para garantizar la interoperabilidad de la información geográfica”.

“La información geográfica gestionada por una IDE se presenta en diferentes formatos; ortofotos, imágenes de satélite, mapas, nombres geográficos, capas SIG, entre otros” (Bermejo y Conti, s.f., 2).

De acuerdo con Nebert (2004) citado por Cedeño, Mondragón, Moraga y Solano (2017, p.2), una IDE no incluye solamente datos y atributos geográficos, sino también “documentación suficiente (los denominados metadatos), un medio para descubrir, visualizar y valorar los datos (catálogos y cartografía en red) y algún método para proporcionar acceso a los datos geográficos (generalmente, Internet)”. Este tipo de interacción de la IDE con el usuario se muestra en la Figura 1.

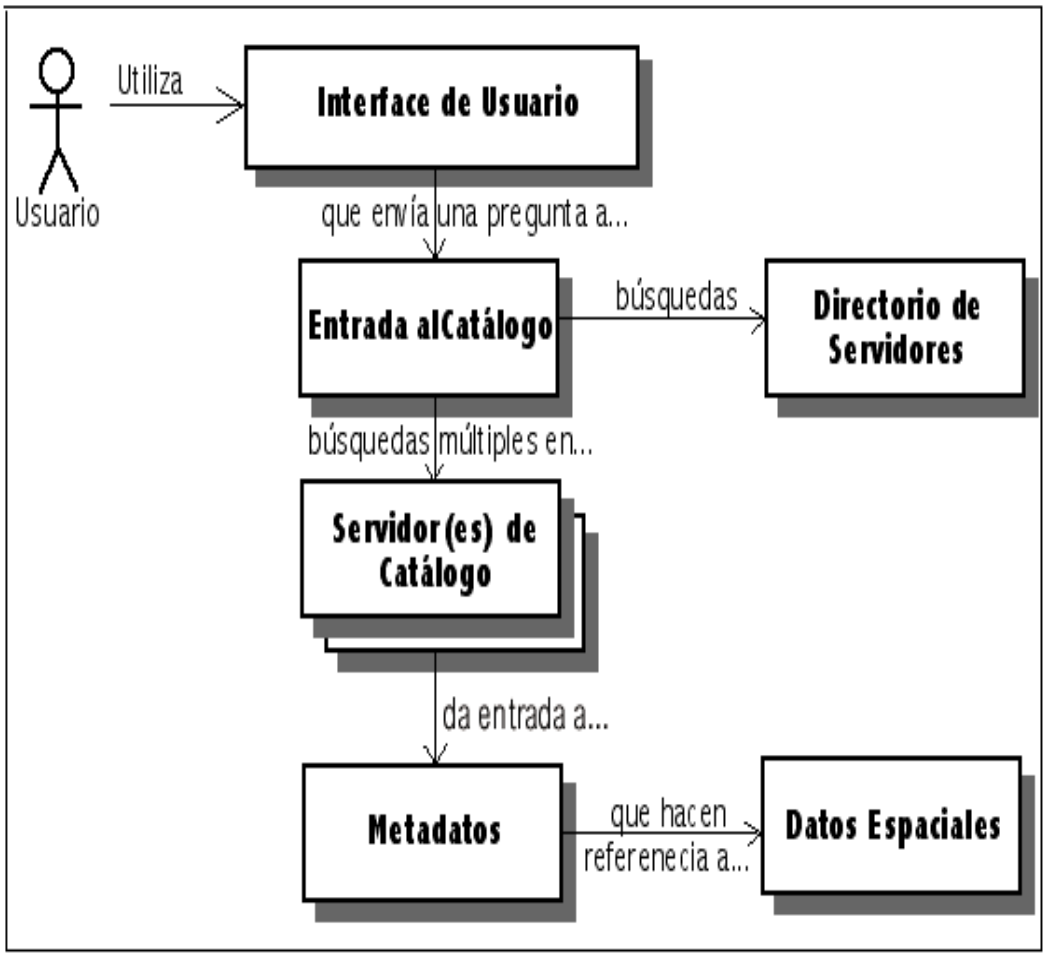

Figura 1: Interacción con servicios de catálogo distribuido

Fuente: RedIRIS, 2001

2.2.1. Interoperabilidad

Desde el punto de vista informático, la interoperabilidad se describe como la capacidad técnica de distintos programas de intercambiar datos a través de un conjunto de formatos destinados para tal cambio.

Según Bernabé y López (2012), la interoperabilidad se puede abordar desde varias perspectivas y varían de un autor a otro, pero aparte de eso, la implementación es un verdadero desafío todavía.

Según la Organización de los Estados Americanos (OEA, 2014), citada por Mesa (2014), algunas acepciones de interoperabilidad pueden ser las que se describen a continuación:

Interoperabilidad Técnica: responde a lo relacionado con *hardware, software* y telecomunicaciones para el fin de interconectar sistemas y servicios. **Interoperabilidad Semántica o informacional**: se hace con el fin de garantizar que la información intercambiada sea precisa y se comprenda adecuadamente. **Interoperabilidad Organizacional**: busca definir los objetivos de negocio y facilitar la colaboración de las entidades involucradas.

Interoperabilidad Político Legal: relacionada con los lineamientos, planes estratégicos y marco legal que permite el intercambio de información **Interoperabilidad Socio Cultural**: se refiere a la cooperación de las administraciones públicas y de otros actores.

Ahora bien, de acuerdo con Bernabé y López (2012), la interoperabilidad de las IDE en el ámbito geográfico busca agilizar el acceso a la información y optimizar la toma de decisiones. Ello se trata de llevar a cabo mediante marcos normativos, que aunque son necesarios, no suelen ser suficientes para lograr la efectiva a nivel de territorio.

2.2.2. Componentes básicos de una IDE

Estándares: los estándares son normas a diferentes niveles. Como los son por ejemplo los internacionales, que se usan para asegurar la transacción eficaz de la información, sin restricciones en la tecnología. Ellos contribuyen a la interoperabilidad tanto técnica (lenguajes, formatos y servicios) como también semántica (coherente en cuanto al significado). Para ello los organismos deben llegar a acuerdos sobre los modelos conceptuales en la información.

Información y datos: Se trata de los datos de las IDE, pero organizados de manera coherente para facilitar su búsqueda y acceso, a fin de facilitar el análisis y visualización. Pero, además de los datos espaciales, también hace parten de la información los servicios de catálogo que permiten descubrir esa información, así como los datos de los datos, es decir, lo metadatos; los cuales, según Bermejo y Conti (s.f.), son otro pilar fundamental ya que describen a los propios datos espaciales y aportan información adicional acerca de ellos. Resultando

imprescindibles para conocer siempre qué cartografía se está usando, y son el único medio del que se dispondrá para poder seleccionar qué cartografía se ajusta mejor a cada caso de uso.

Además de ello, es igualmente importante contar con las herramientas para encontrar los datos en un conjunto variable y normalmente muy grande de información. Esto medios se conocen como catálogos, y permiten encontrar o seleccionar la información de interés.

Acuerdos: estos se llevan a cabo entre los distintos participantes de las IDE, principalmente los proveedores de la información, los gobiernos y la academia, como líderes en el proceso de la disponibilidad de la información, a fin de establecer servicios y tecnología compatible para el intercambio de información y puesta a disposición de los datos para el usuario final.

Políticas: el sector público es generalmente el mayor recolector de información, y por lo tanto, su disposición para garantizar el mantenimiento, organización y disponibilidad de la información resulta fundamental. Al tiempo que su intervención puede ayudar a conciliar el interés del sector privado por la información que el estado guarda. La intermediación ayuda a que se intercambien datos entre el sector público y privado en favor de los usuarios de las IDE.

Redes accesibles: la accesibilidad de la información geográfica se materializa, en el caso de las IDE, a través de Internet. De manera que la conectividad debe ser la adecuada para que pueda soportar las peticiones de los usuarios al servidor. Muchas veces esas peticiones son múltiples y la tecnología debe estar preparada para esos casos.

Usuarios: los esfuerzos institucionales de las IDE deben estar encaminados a cubrir las necesidades de sus usuarios, y facilitar las herramientas para el máximo aprovechamiento de la información disponible, a fin de que su uso se garantice. Para ello es necesario realizar una adecuada caracterización del usuario y sus necesidades

Todos estos elementos descritos se presentan de manera gráfica en la Figura 2.

Figura 2: Componentes de una IDE
Fuente: SNIT Costa Rica (s.f.).

2.2.3. Actores de una IDE

Según Mascarell (2014), entre los actores que se puede encontrar en las IDE está en primer lugar el segmento de los usuarios que se representan por diferentes actores, como lo es por ejemplo el mismo gobierno, que requiere información para la gestión pública; así como también las universidades, los centros de investigación y finalmente las empresas privadas y personas individuales. A medida que mejor se contemplen las necesidades de los diferentes tipos de usuario, mayor será el éxito de la IDE como institución referente en la consulta de información geográfica.

De forma más detallada, Portolés y Martínez (2005), dividen los tipos de usuario según su perfil, de la siguiente manera:

Usuario básico. Utiliza las herramientas básicas para la visualización y/o descarga de la información.

Usuario avanzado. Utiliza herramientas y aplicaciones no disponibles para el público general, ya sea a través de la web o mediante aplicaciones locales.

Usuario de negocio. Accede a los datos de la IDE desde aplicaciones externas, para combinarlos con otros fuera de la IDE y realizar algún tipo de negocio.

Usuario consultor. Está autorizado a acceder a datos restringidos de una temática específica.

Usuario editor. Encargado de mantener un subconjunto de datos existentes en la IDE.

Usuario gestor. Gestiona determinados servicios proporcionados por la IDE, por ejemplo un servicio de mapas temáticos concreto.

Administrador. El responsable final de mantener la infraestructura y dar soporte técnico a los restantes usuarios.

A medida que se desciende en esta clasificación, aumenta la especialización, al tiempo que disminuye el número de usuarios que pertenecen a cada una de las clases definidas. Tal y como se recoge esquemáticamente en la Figura 3*Figura 3*.

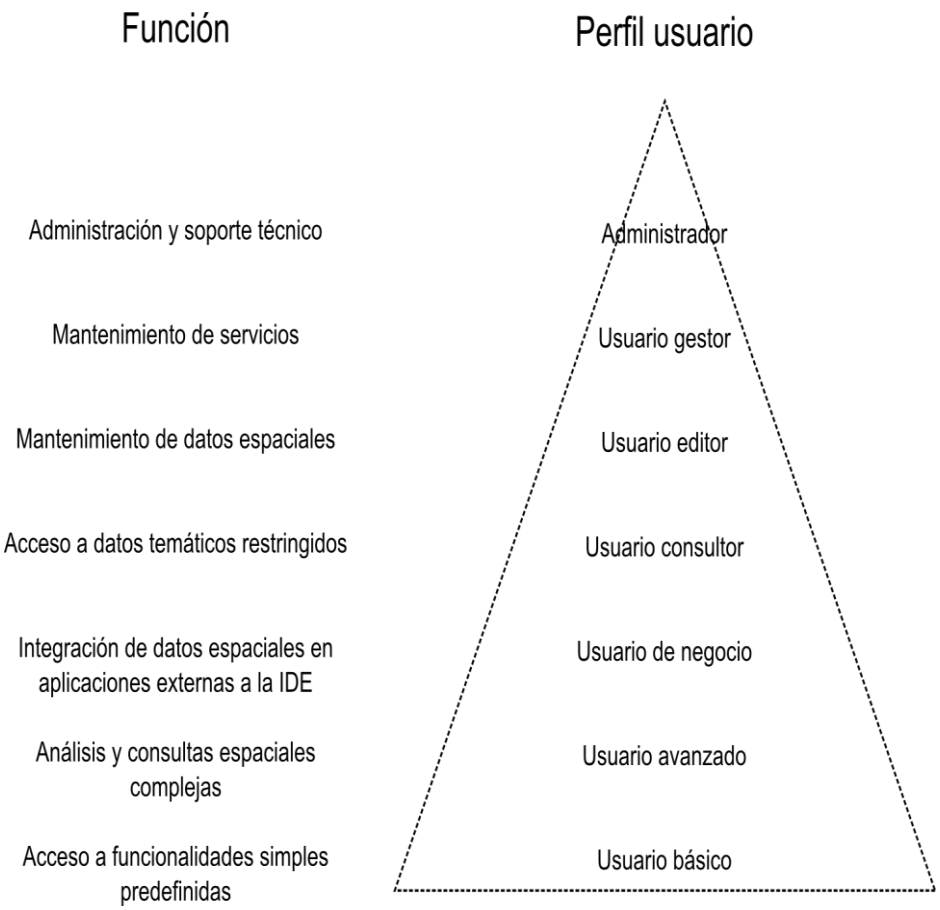

Figura 3: Clasificación de usuarios basada en perfiles
Fuente: Portolés y Martínez (2005)

2.2.4. Niveles de una IDE

Para Mascarell (2014. p.13) “Una IDE está formada por un conjunto de nodos, cada uno de ellos administrado por un organismo responsable. Los nodos de una IDE se encuadran en distintos niveles en función del tipo de organismo responsable del nodo, el cual se clasifica en función de la posición en la jerarquía administrativa que ocupe el organismo responsable del nodo, por niveles, desde la IDE global que está en el nivel superior hasta las IDE locales y las IDE corporativas en el nivel inferior. El detalle de la información geográfica crece según se desciende en esta jerarquía, de la misma manera que el área geográfica cubierta por la IDE también disminuye.”.

La Figura 4 ilustra estos niveles:

Figura 4: Niveles de jerarquía de una IDE
Fuente: SGC, 2017.

IDE Global

Para el caso de la IDE global, el nivel de detalle de la información no es muy alto (más bien es de poco detalle, por lo que el orden de la escala es alrededor de 1:1’000.000), sin embargo los lineamientos en cuanto a estándares de interoperabilidad pueden cubrir a las demás IDE que están en los niveles base de la pirámide. Un ejemplo de IDE global es la Asociación Global de Infraestructuras de Datos Espaciales (Global Spatial Data Infrastructure Association GSDI-

http://gsdiassociation.org/) creada en 1996, y que según el portal IDECA (s.f., ¶ 1) “promueve la cooperación internacional y la colaboración en apoyo de los entes locales, nacionales e internacionales referente al manejo de las IDE”.

IDE Nacional

A nivel nacional las entidades gubernamentales también utilizan información con baja resolución o a pequeña escala, sin embargo es usual que se requiera mayor detalle en la información para proyectos de importancia nacional, y para ello se debe contar con la cooperación tanto entre las entidades del estado como también entre los gobiernos locales.

En dicho sentido toman especial relevancia las IDE nacionales, para centralizar el acceso a la información espacial y no duplicar esfuerzos en la creación de la información, pero también para asegurar la calidad de la misma (Bernabé y López, 2012). Además, este proceso de establecimiento de una IDE nacional ayuda a crear y usar políticas de acceso a los datos según las necesidades y el contexto propio del país, teniendo en cuenta los estándares internacionales adaptados a nivel país para garantizar la interoperabilidad de la información.

IDE Regional

La IDE regional consiste en dos o más países organizados a fin de crear lineamientos y compartir información geográfica de interés común. A nivel de las américas se tiene como referencia al Comité Permanente para la Infraestructura de Datos de las Américas - CP IDEA, que según Hansen (2006), promueve el uso de las IDE como instrumentos de gestión y administración de la información espacial, para atender temas de desarrollo social y económico.

IDE Local

Los requerimientos de información geográfica a nivel local son más demandantes en términos de nivel de detalle. Según Bernabé y López (2012, p. 335), dicha información consiste en “cartografía de base, transporte, aspectos sociales y culturales, ordenación del territorio, medio ambiente, servicios públicos,

entre otros". Un ejemplo de IDE local es la IDESC, justamente, que agrupa a instituciones de orden público y privado.

IDE corporativa

Las IDE corporativas o institucionales "articulan la información que gestiona una empresa, instituto u organismo hacia el interior de la organización, haciendo más eficiente su uso por múltiples usuarios." (Bernabé y López, 2012, p. 335). En dicho sentido, las IDE corporativas producen y ponen a disposición información geográfica según sus propósitos misionales, pero en concordancia con las IDE nacionales. Tal es el caso de la IDE Geocientífica del Servicio Geológico Colombia (SGC), que busca beneficiar a los actores del sector minero energético y además al sector público y privado interesado en el tema (SGC, s.f.).

2.2.5. Geoportal IDESC. Cali, Colombia

Se puede definir un Geoportal como un punto de acceso a la información geográfica, utilizando como vía de acceso el internet.
En general los datos que ofrece un Geoportal pueden ser de lo más variado, permitiendo la visualización de los datos y favoreciendo la integración, interoperabilidad e intercambio de información entre los usuarios y las instituciones.

Según la IDESC (2016), su Geovisor es una herramienta facilitadora de la información normativa y cartográfica del municipio, que busca armonizar los procesos de captura y acceso de la información producida por sus entidades asociadas. Es de aclarar que la IDESC cumple un papel de facilitadora, y por lo tanto el mismo portal advierte lo siguiente: "en caso de dudas, inquietudes o posibles imprecisiones así como para la expedición de conceptos normativos y documentos oficiales, es necesario utilizar y citar la información de la fuente oficial" (IDESC, 2016, ¶ 5).

2.3. OPEN GEOSPATIAL CONSORTIUM (OGC)

A partir de su creación, en 1994, el Open Geospatial Consortium (OGC) ha impulsado el posicionamiento de la información geográfica como una parte integral de la infraestructura mundial de información (OSGeo Live, 2011).
Su fin es la definición de estándares abiertos en el campo de los Sistemas de Información Geográfica, mediante acuerdos que faciliten el efectivo intercambio de los datos geoespaciales (IDECA, s.f.).

2.3.1. ESTÁNDARES OGC

Una IDE debe posibilitar acceder a los datos geográficos contenidos en ella, desde distintos puntos de forma simple y eficaz, a través de estándares de interoperabilidad, los cuales se encargan de aportar la homogeneidad necesaria para hacer esto posible.

De acuerdo a esto, según la IDE Canarias (2015), los estándares más importantes surgidos del OCG son:

Geography Markup Language (GML)

Grid Coverages (GC)

Web Coverage Service (WCS)

Web Feature Service (WFS)

Web Map Context Documents (WMC)

Styled Layer Descriptor (SLD)

Web Map Service (WMS)

Dichos estándares, según Bernabé y López (2012, p. 268), “son documentos que detallan interfaces informáticas o formas de codificación de datos. Su implementación en productos y servicios debe producir resultados independientes del productor y del sistema, además de permitir que los desarrollos informáticos

puedan intercambiar datos sin necesidad de adecuar los correspondientes códigos en el lado del cliente o del servidor".

Pero además, según Olaya (2014, p. 782), "la interoperabilidad implica que se puede sustituir unos elementos del sistema (como por ejemplo los clientes y servidores) por otros distintos, teniendo la seguridad de que van a interactuar entre ellos sin dificultades." Como ejemplo práctico para el caso de los SIG, Olaya (2014) afirma que el formato *shapefile* para capas vectoriales, es uno de estos estándares implícitos ya que está ampliamente difundido en el mundo SIG.

Para que los estándares se puedan materializar de manera efectiva deben ser abiertos, con unos principios fundamentales que según Olaya (2014) son:

Disponibilidad. Los estándares abiertos deben estar disponibles para la lectura y uso en cualquier implementación.

Máxima posibilidad de elección. Utilizando estándares abiertos, la competencia entre fabricantes ha de basarse puramente en las capacidades que ofrecen, con lo que los consumidores ganan en calidad de los productos y en posibilidades de elección, creando un mercado competitivo y justo.

Gratuidad. Implementar un estándar es gratuito, sin necesidad de pagar, contrario a lo que sucede en el caso de una patente por ejemplo.

Discriminación. Los estándares abiertos y las organizaciones productoras de los mismos, no se inclinan a favor de uno u otro implementador para favorecerlo

Extensión o creación de subconjuntos de un estándar. Los estándares abiertos pueden ser extendidos o bien, presentados como subconjuntos del estándar original.

Prácticas predatorias. Los estándares abiertos no pueden prohibir el desarrollo de extensiones de los estándares, pero pueden requerir que se publique toda la información referente a ese desarrollo hecho.

Según Olaya (2014), para maximizar el valor de la IDE es imprescindible que el acceso a los datos geográficos no presente problemas. Para ello es importante definir de forma adecuada cómo se establece la comunicación entre clientes y servidores, de forma que estos primeros no solo puedan obtener los propios datos geográficos de estos últimos, sino también realizar consultas o conocer qué otras funcionalidades se encuentran disponibles.

De manera que los estándares abiertos son fundamentales en el proceso de la interoperabilidad. Esto se refleja de manera esquemática en la Figura 5*Figura 5*:

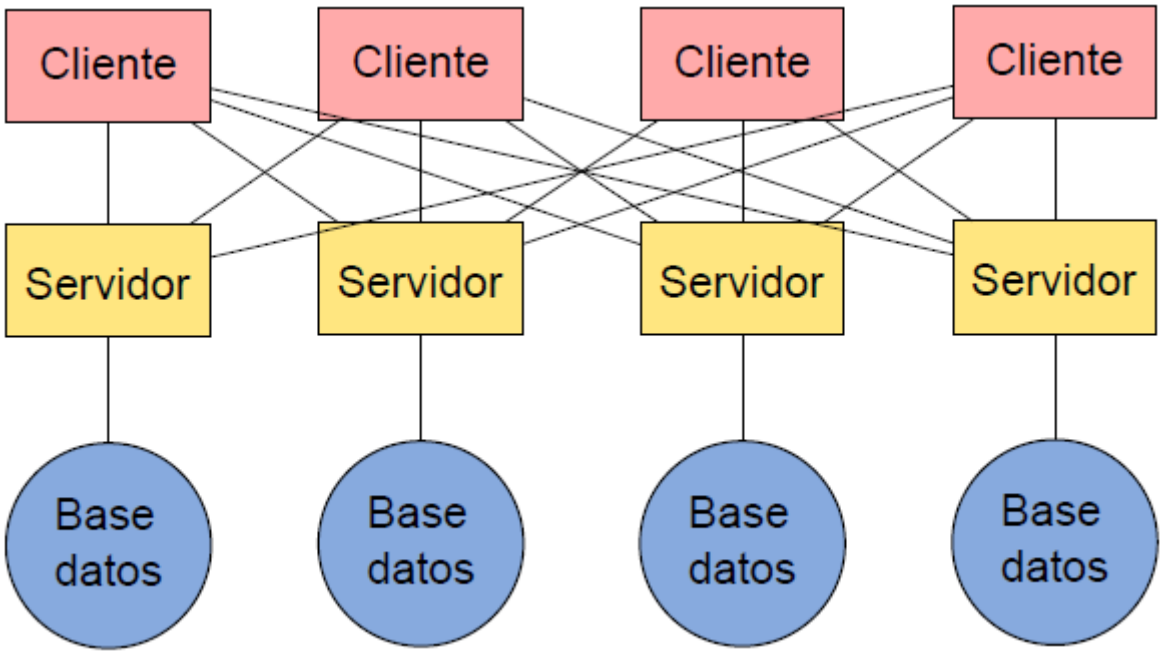

Figura 5: Esquema ideal de una arquitectura interoperable.
Fuente Olaya (2014).

Como se puede ver en la Figura 5, “existe un servidor que es el que gestiona y ofrece los servicios para cada base de datos, pero a él pueden acceder todos los clientes, ya que por el hecho de estar basados en estándares abiertos es posible una comunicación plena entre cualquiera de ellos” (Olaya, 2014, p. 785)

2.3.2. WFS

De acuerdo con Bernabé y López (2012), el objetivo principal del servicio WFS es brindar las condiciones técnicas y operativas para poder publicar, acceder, consultar y descargar información geográfica en formato nativo y editable.
Hoy en día es posible consultar y descargar los datos almacenados en servidores remotos, mediante la especificación WFS a través de programas SIG, en un formato editable.
Las operaciones que soporta este tipo de servicio son las siguientes:

GetCapabilities: “Es una operación que permite recuperar el fichero de capacidades, de tal manera que se puede conocer el nombre del servicio, el estándar OGC que cumple, la versión, quién publica el servicio, la zona que cubre, las restricciones de uso, un punto de contacto, el sistema de referencia, las capas que incluye, entre otros” (Bernabé y López, 2012, p. 161).

DescribeFeatureType: Describe la estructura de cualquier tipo de geometría que puede ser servida (consultada o descargada).

GetFeature: Devuelve la instancia de una geometría de acuerdo a las consultas realizadas, donde se pueden solicitar todo el conjunto de datos capa a capa, o también mediante filtros de atributos con el lenguaje *Common Query Language* (CQL).

Según la Junta de Andalucía (s.f), existe también otras operaciones de tipo transaccional (WFS-T) para los casos donde dichos servicios están habilitados, que es generalmente para grupos de trabajo internos. Estos son: *Transaction* y *LockFeature*, y se describen a continuación:

Transaction: Permite realizar operaciones de edición (crear, eliminar o modificar elementos)

LockFeature: Permite el bloqueo de una o varias capas mientras está teniendo lugar una operación transaccional.

GetGMLObject: Permite obtener *features* y elementos por el ID de un servicio WFS. Dado el ID de un elemento devuelve el GML que describe dicho elemento. (Junta de Andalucía, s.f.)

2.3.3. GetCapabilities

La operación de GetCapabilities (consulta de capacidades) “devuelve un fichero XML con información sobre el servicio WFS, como por ejemplo el autor o proveedor, los sistemas de referencia que soporta, los formatos de salida, los objetos geográficos que contiene, operaciones que soporta para cada tipo de entidad.” (IDEE, 2018, ¶ 7). De manera concreta, contiene la información suficiente para identificar plenamente el servicio y la información contenida en él, bajo los pilares que se pueden ver en la Figura 6:

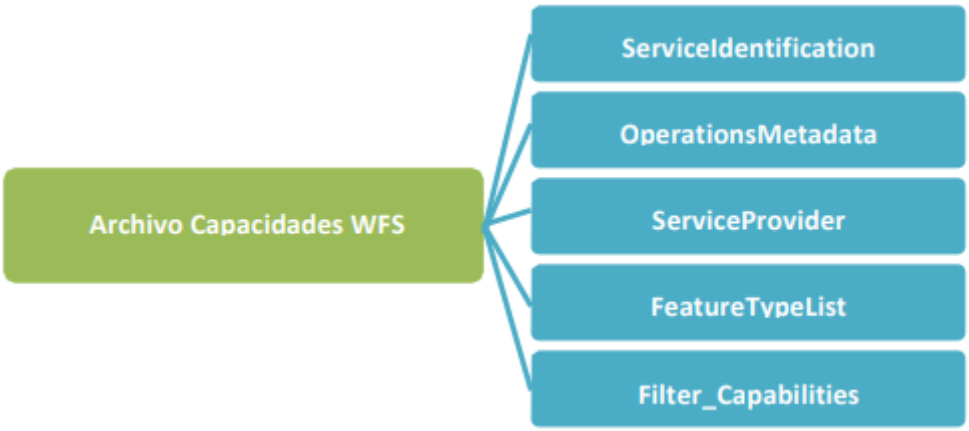

Figura 6: Contenido del fichero XML, de respuesta a la consulta GetCapabilities
Fuente: IDEE (2018).

En la práctica, la petición GetCapabilities se hace efectiva copiando y pegando el URL del servicio WFS en el navegador de Internet, y agregando los parámetros siguientes: SERVICE = WFS, REQUEST = GetCapabilities, VERSIÓN = 1.1.0. Al ingresar esos elementos en el navegador de Internet se deben separar por el símbolo **&**. El parámetro VERSION es opcional, mientras los demás son obligatorios (IDEE, 2018).

2.3.4. GetCapabilities y Web Feature Service (WFS)

Una solicitud *GetCapabilities* utiliza el formato codificado de pares clave-valor a través de una solicitud GET de HTTP (IBM, s.f.). A continuación, en la tabla 1, se presentan los elementos clave de la solicitud *GetCapabilities*, así como también si su uso es obligatorio u opcional.

Tabla 1. *Claves de GetCapabilities*
Fuente IBM, s.f.

Clave	**Obligatorio u opcional**	**Definición y ejemplo**
service	Obligatorio	Identificador del tipo de servicio. *service*=WFS
request	Obligatorio	Nombre de la operación *request=GetCapabilities*
AcceptVersions	Opcional. Devuelve la versión soportada más reciente, si se ha omitido.	Secuencia priorizada separada por comas de una o más versiones de especificaciones aceptadas por el cliente, con las versiones preferidas listadas primero. *AcceptVersions* =1.1.0,1.0,0
updateSequence	Opcional. Devuelve la versión más reciente del documento de metadatos si se ha omitido o si el servidor Web no la soporta.	Versión de servicio del documento de metadatos. El valor aumenta cada vez que se realiza algún cambio en el documento de metadatos completo. updateSequence=123

Clave	Obligatorio u opcional	Definición y ejemplo
AcceptFormats	Opcional. Devuelve un documento de metadatos de servicio que utiliza los tipos MIME text/xmlsi se omite o si el servidor Web no lo soporta.	Secuencia separada por comas de cero o más formatos de respuesta para el cliente. Lista los formatos preferidos primero. *AcceptFormats=text/xml*

2.4. PYTHON

El lenguaje de programación Python fue creado por Guido van Rossum en 1989, y poco a poco se ha dado a conocer como un lenguaje de scripts o "pseudocódigo ejecutable", con una sintaxis simple y alta legibilidad. Siendo reconocido popularmente como un lenguaje idóneo para la introducción a la programación (García, 2017).

2.4.1. Características de Python

Python es un lenguaje de programación de alto nivel, multiplataforma, de tipado dinámico y multiparadigma, con reglas de estilos para escribir el código de manera estandarizada; que tiene además sus propios módulos que ayudan a realizar muchas tareas comunes sin necesidad de tener que programarlas desde cero; también es posible importar módulos que no estén incluidos en su librería de módulos estándar (Bahit, 2012).

"Python es liberado bajo una licencia propia llamada Python *License* que ha sido certificada por el movimiento *Open Source*, y es compatible con la GPL (GNU *Public License*) de la *Free Software Foundation* (Fundación del *Software* Libre)" (Challenger, Díaz y Becerra; 2014, p. 7). Por lo cual se puede usar tanto para hacer *software* libre como *software* privativo.

Según Bahit (2012), Python es un lenguaje con las siguientes características:

De alto nivel: lo que significa que tiene una estructura sintáctica y semántica legible, acorde a las capacidades cognitivas humanas.

Interpretado: a diferencia de otros lenguajes, Python no requiere de un compilador para ser ejecutado sino solamente un intérprete. Un intérprete ejecuta el programa directamente, sin que se requiera crear un ejecutable.

Tipado dinámico: significa que las variables usadas en el código no requieren ser definidas asignando su tipo de datos, sino que se auto-asigna en tiempo de ejecución, según el valor declarado.

Multiplataforma: significa que puede ser interpretado en diversos Sistemas Operativos, siempre y cuando exista un intérprete programado para su ejecución.

Multiparadigma: acepta diferentes técnicas de programación, tales como la orientación a objetos así como también programación imperativa y funcional.

Interactivo: a través de su intérprete por medio de línea comandos, es posible introducir sentencias, ejecutarlas y mostrar resultados visibles (DesarrolloWeb, 2003).

Orientado a objetos: la programación orientada a objetos (POO, u OOP por sus siglas en inglés) es un paradigma de programación que tiene por objetivo organizar el código fuente, y re-usarlo en similares contextos (Covantec, 2014). Según Covantec (2014), en el caso de Python los objetos son la abstracción para data. Toda la data en un programa Python es representado por objetos o por relaciones entre objetos. Cada objeto tiene una identidad, un tipo y un valor.

Sintaxis: Python tiene una sintaxis muy clara y visual, debido a su notación indentada, es decir código tabulado o con margen hacia el interior del cuerpo del código para agrupar porciones del mismo. Esto es útil para tener una mejor legibilidad y distinguir de manera clara las funciones y ciclos (bucles) (DesarrolloWeb, 2003).

Librería Estándar: para Challenger, Díaz y Becerra (2014); una de las fortalezas de Python es la librería estándar con que cuenta. Con decenas de módulos que cubren la mayoría de las necesidades básicas de un programador. Con tópicos como:

- Cadenas
- Estructura de datos
- Funciones numéricas y matemáticas
- Compresión de datos
- Formatos de archivo
- Criptografía
- Servicios de los Sistemas Operativos
- Comunicación entre Procesos
- Manejo de datos de Internet
- Servicios multimedia
- Manejo de excepciones

Se puede afirmar que es una de las más completas con que se cuenta en la actualidad, comparable con la de Java y .NET.

Rendimiento: aunque Python es un lenguaje interpretado y estos tienden a ser más lentos que los lenguajes compilados, su librería estándar está implementada en el lenguaje C, lo que hace que sus funciones primitivas sean bastante eficientes; además, puede compilarse su código a bytecodes, similar al que usan Java y .NET, lo que optimiza aún más el proceso de interpretación (Challenger, Díaz y Becerra; 2014).

Documentación: otra de las características principales de Python es la inclusión de un sistema de documentación desde el diseño del lenguaje. Estas cadenas de documentación pueden llamarse inclusive en tiempo de ejecución, por lo que se pueden consultar en el propio intérprete usando la función *help* (Challenger, Díaz y Becerra; 2014)

Extensibilidad: Python puede reutilizar código escrito en los lenguajes C y C++, permitiéndole aprovechar las bases de código de esos lenguajes.

En cuanto a la tarea de desarrollar como tal, existen herramientas como son los Entornos de Desarrollo Integrado (IDE o IDLE- *Integrated DeveLopment Environment)*, que tienen diversas características y complejidad. Actualmente es posible incluso encontrar portales web para desarrollar y ejecutar código Python de manera rápida. Hay que anotar que los resultados, como lo es por ejemplo la descarga de datos, quedan almacenados en el servidor de esos IDLE en línea.

2.4.2. Aplicaciones prácticas de Python

Python es un lenguaje bastante fácil de aprender en comparación con otros de su mismo nivel, por lo que es una de las primeras opciones a la hora escoger una herramienta para el tratamiento de datos, como por ejemplo de tipo científico o también de tipo espacial en conjunto con librerías complementarias desarrolladas especialmente para ese propósito.

Python en la ciencia: la ciencia se vale de herramientas como Python, por ser simple y eficiente. Por lo que se puede encontrar el uso del lenguaje en Bioinformática, Neurofisiología, Física, Matemáticas, entre otros. A tal punto que grandes centros de investigación como el CERN (Organización Europea para la Investigación Nuclear) lo han adoptado por su poder en el tratamiento de datos (Challenger, Díaz y Becerra, 2014).

Python y el geoprocesamiento: Una herramienta típica de geoprocesamiento toma los datos de entrada, de formatos como por ejemplo *feature class*, raster, o tabla y produce datos de salida como resultado. Este tipo de proceso soporta la automatización de flujos de trabajo mediante la creación de una secuencia que combina una serie de herramientas de tal manera que la salida de una herramienta se convierta en la entrada de la siguiente. Estos flujos de trabajo automatizados que combinan herramientas de geoprocesamiento se puede lograr con programas de SIG o haciendo uso de la programación en lenguajes como Python a través de scripts o desarrollos más complejos (Díaz, 2016).

2.5. BIG DATA

El término *Big Data* o Datos Masivos, según expone Snijders, Matzat, y Reips (2012), se refiere a colecciones de datos que son tan grandes o complejos que no se pueden tratar como conjuntos de datos tradicionales. Con el avance de la tecnología, datos de comunicación entre máquinas, datos producidos por los usuarios en la web 2.0, datos acumulados por diversas industrias o centros de investigación, ha surgido una nueva problemática, ya que no solo está el problema del gran volumen de datos, sino que también estos datos pueden ser extremadamente variados entre sí, y la construcción de aplicaciones que analicen estos datos a una velocidad adecuada, o los capturen, busquen, compartan y almacenen no se pueden realizar con tecnologías tradicionales como las bases de datos convencionales.

En resumen, y de acuerdo con Pascual (2017), se puede hablar de Big Data cuando se cuenta con varios Terabytes de información, que pueden o no estar organizados u ordenados. Por lo tanto se requiere aplicar una serie de técnicas y dominar conocimientos que hacen parte de lo que se conoce como *Data Science* (ciencia de datos).

2.6. DATA SCIENCE

Almacenar y procesar gran cantidad de información requiere de un conjunto de herramientas tecnológicas y conocimientos de matemática, estadística e inteligencia de negocio de manera tal que se pueda manejar grandes proporciones de datos, tanto en cantidad como en calidad. En ese entendido aparece la *Data Science* (Ciencia de Datos), que cuenta con diferentes técnicas y modelos para procesar la información, entre lo que se puede mencionar: analítica descriptiva, estadística, *data mining* y el *machine learning* (Pascual, 2017). Las relaciones entre estos conceptos mencionados los podemos ver en la Figura 7*Figura 7*:

Figura 7: Data Science y los conocimientos requeridos
Fuente: Lages (2018)

La utilidad de la información con la que se cuente para trabajar depende mucho de la calidad de los datos de entrenamiento, pues esta debe ser abundante, pero más que eso debería representar una porción significativa de datos del problema y su complejidad, para que brinden una idea del grado de facilidad con el que un conjunto de datos puede ser aprendido (González, Gómez, Pastrana y Hernández, 2014).

La ciencia de datos se relaciona de manera tangencial con otro concepto incluso más amplio que es la Inteligencia Artificial, tal como se observa en la*Figura 8* Figura 8:

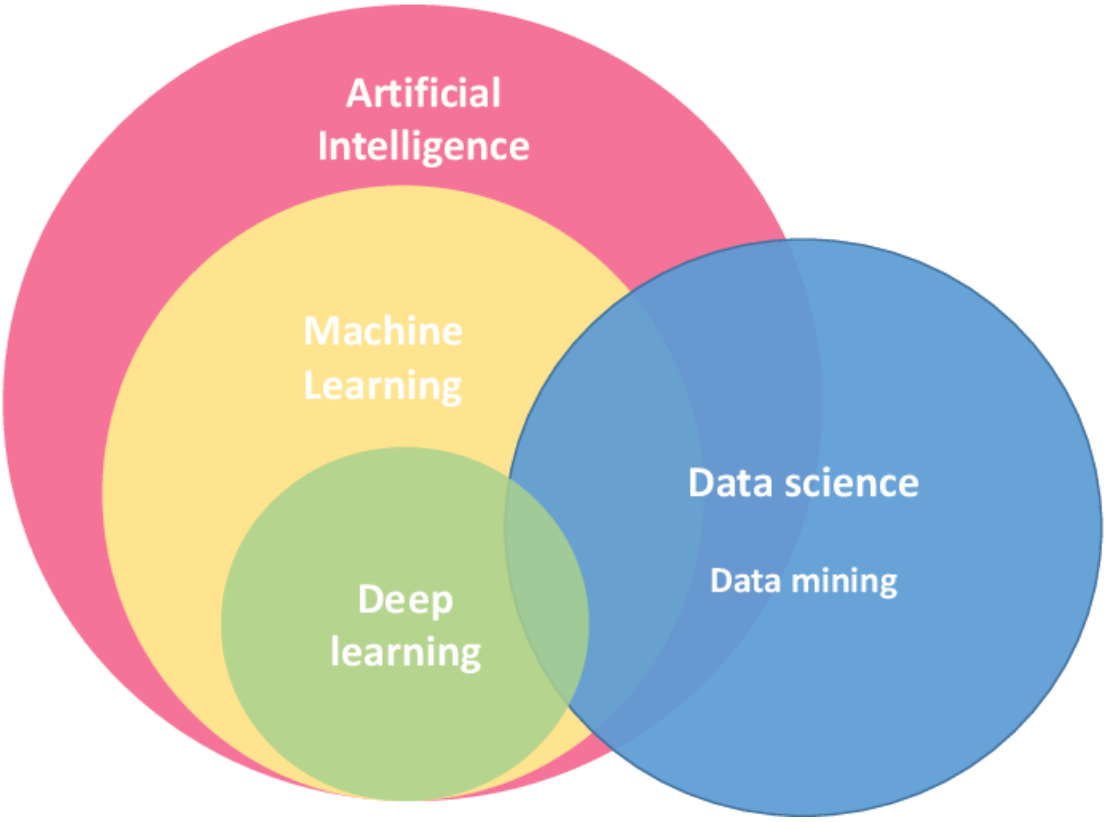

Figura 8: Relación Ciencia de Datos e Inteligencia Artificial
Fuente: Kulin, Kazaz, Moerman y Poorter (2020)

2.6.1. Data Mining

Según Piatetsky et al. (1992), citados por Matos, Chalmeta y Coltell (2006, p. 18), la definición formal de *data mining* o minería de datos es, "el conjunto de técnicas y herramientas aplicadas al proceso no trivial de extraer y presentar conocimiento implícito, previamente desconocido, potencialmente útil y humanamente comprensible, a partir de grandes conjuntos de datos, con objeto de predecir de forma automatizada tendencias y comportamientos previamente desconocidos".

La minería de datos hace parte de la ciencia de datos, y es una herramienta para explorar, extraer y exponer información útil, que de otro modo sería difícil de reconocer. Los métodos asociados a cada concepto se pueden apreciar en la Figura 9:

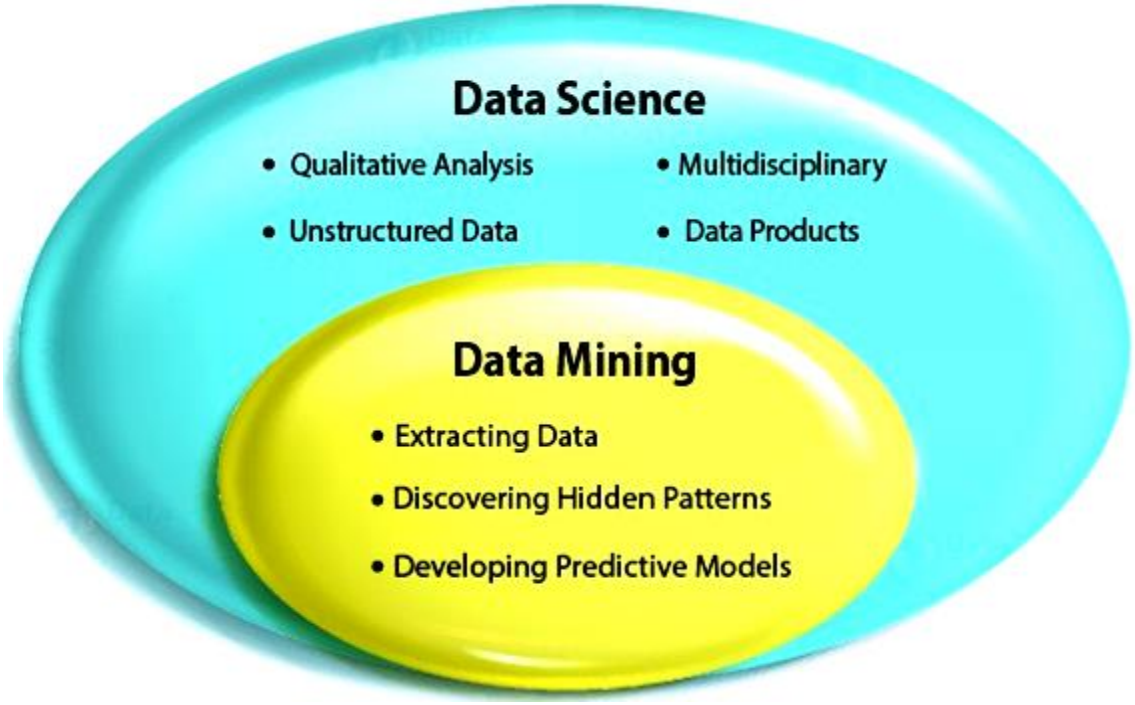

Figura 9: Relación entre Ciencia de Datos y Minería de Datos
Fuente: Gour (2019)

De acuerdo con Mata (2017), citado por Vallejo, Guevara y Medina (2017); la minería de datos trata de aprovechar la gran cantidad información a la que se puede acceder hoy en día, y la potencia de los nuevos computadores para realizar operaciones de análisis sobre esos datos. El *Data Mining* permite encontrar información escondida en los datos, mediante un proceso de identificación de información relevante con el objetivo de descubrir patrones y tendencias.

Técnicas de *Data Mining* (minería de datos)

Según Clinic Cloud (s.f.), en el ámbito de la investigación las técnicas de *Data Mining* pueden ayudar a los científicos a clasificar y segmentar datos, además de formar hipótesis, y se puede clasificar en dos tipos:

- Métodos descriptivos: Un método descriptivo es por ejemplo el clustering, que busca descubrir reglas de asociación y patrones secuenciales. Y son utilizados, por ejemplo, para ver qué productos suelen adquirirse conjuntamente en el

supermercado.

• Métodos predictivos: “Usan algunas variables para predecir valores futuros o desconocidos de otras variables. Algunos métodos de este tipo son los siguientes: clasificación, regresión y detección de la desviación. En medicina, por ejemplo, los métodos predictivos pueden emplearse en tareas como clasificar tumores en benignos o malignos.” (Clinic Cloud, s.f., ¶ 7).

Un esquema que representa estos dos métodos y sus ramificaciones se muestra en la Figura 10:

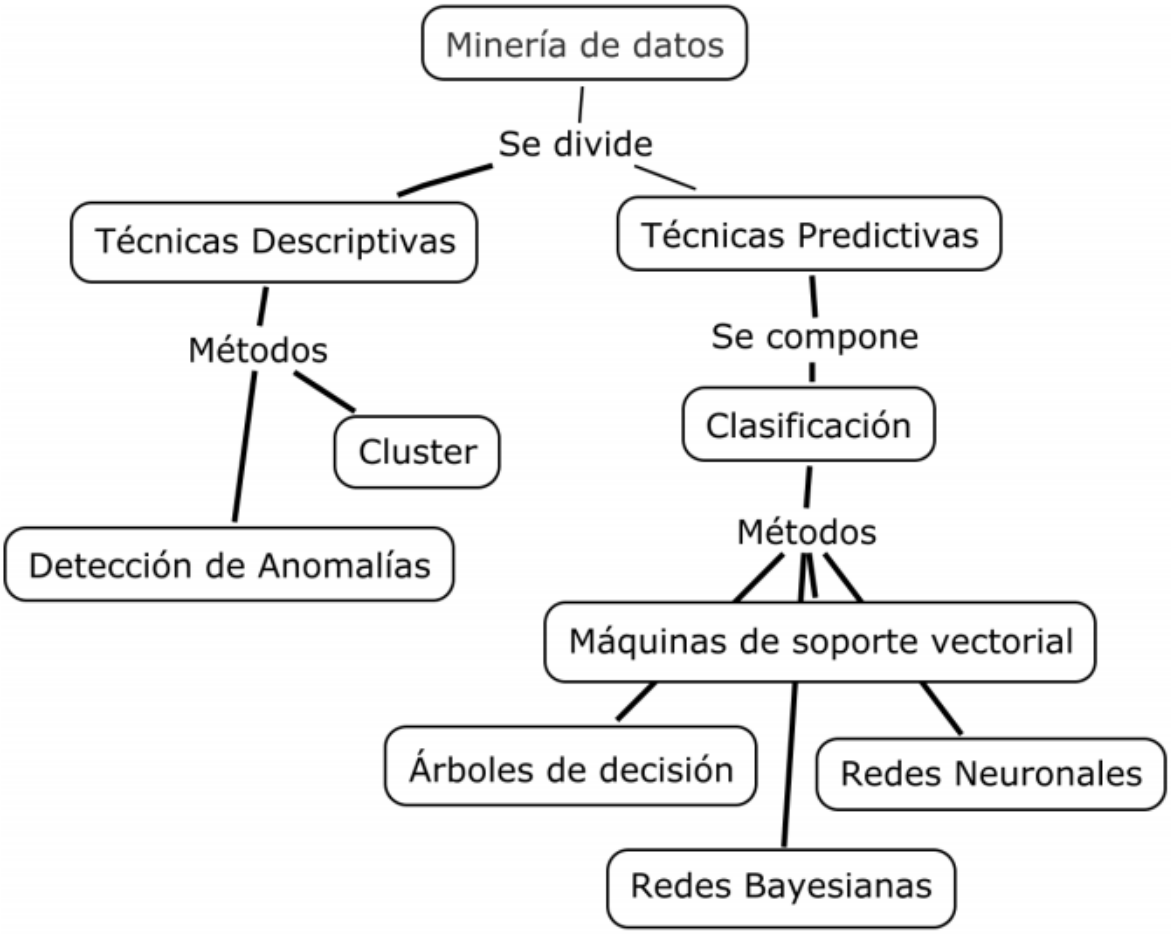

Figura 10: División taxonómica del *Data Mining*

Fuente: Santamaria (2006)

Según Clinic Cloud (s.f.), el proceso de *data mining* tiene su propio estándar, el CRISP-DM (*Cross-Industry Standard Process for Data Mining*), que establece seis pasos a seguir para aplicar el proceso:

- Entender el área en el que se quiere usar *data mining* para definir con claridad el problema.
- Recolectar y entender los datos.
- Preparar los datos: hacer tablas con los campos requeridos, eliminar datos innecesarios.
- Seleccionar la técnica de modelado: construcción del modelo y puesta a prueba del modelo.
- Evaluar de los resultados y revisión del proceso.
- Desplegar e implementar un proceso de *Data Mining* repetible.

De manera esquemática se puede representar el proceso como se observa en la Figura 11:

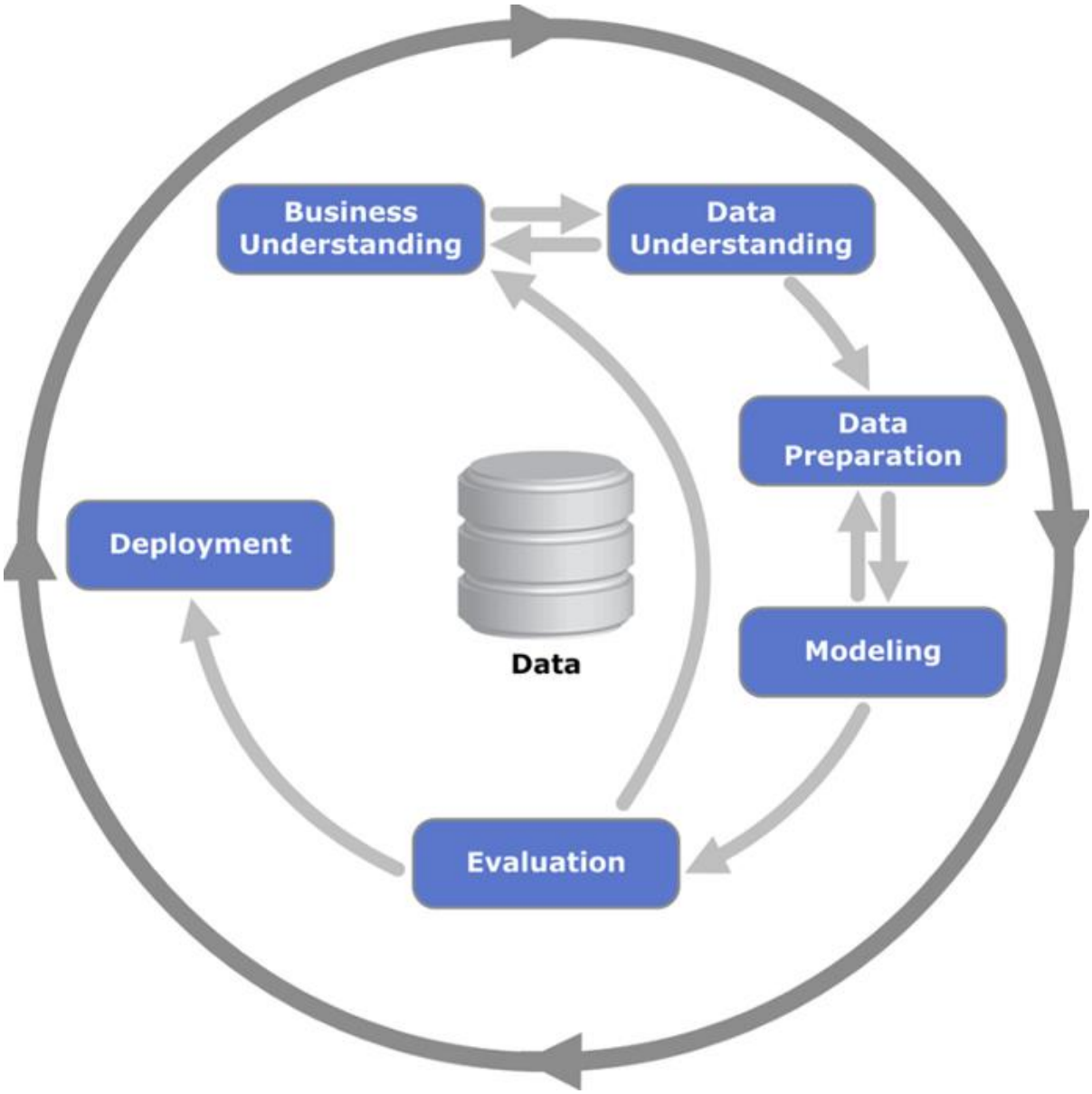

Figura 11: Proceso estándar *Data Mining*.
Fuente: Jensen (2012).

Según Pujari (2001), citado por Ramos (2016), algunas de las principales técnicas de minería de datos basadas en el descubrimiento de información entre los datos son:

- Reglas de asociación: Se usan para descubrir fenómenos en común o relaciones que ocurren en un conjunto de datos. De acuerdo con las reglas de asociación se pueden usar en contextos como el análisis Web, marketing, detección de intrusos, producción continua, bioinformática, entre otros.

- Reglas de clasificación: Basado en el descubrimiento de reglas que permiten “particionar” los datos en conjuntos disjuntos.
- Redes neuronales: Inspirado en la forma en que funciona el sistema nervioso, se construye un sistema de aprendizaje y procesamiento automático en el que un sistema de neuronas compone una red que colaboran entre ellas para producir el resultado de salida. Según González et al. (2014), este tipo de análisis es comúnmente utilizado en las ciencias sociales y tecnológicas: manufactura, biología, finanzas, previsión del tiempo, análisis de tendencias y patrones, entre otros.

2.6.2. Inteligencia Artificial

De acuerdo con Itelligent (2018), la Inteligencia Artificial (IA) tiene como objetivo que a través de la computación y el *Machine Learning* se puedan realizar determinadas operaciones que se consideran propias de la inteligencia humana, llegando incluso a gestionar nuevas situaciones sin la intervención del ser humano. Según Abarca, Campos y Reinoso (2017) la IA se usa para resolver problemas de predicción, clasificación y reconocimiento de patrones. Los temas que abarca la IA de manera directa son el Aprendizaje Automático y el Aprendizaje Profundo, tal como se observa en *Figura 12*Figura 12:

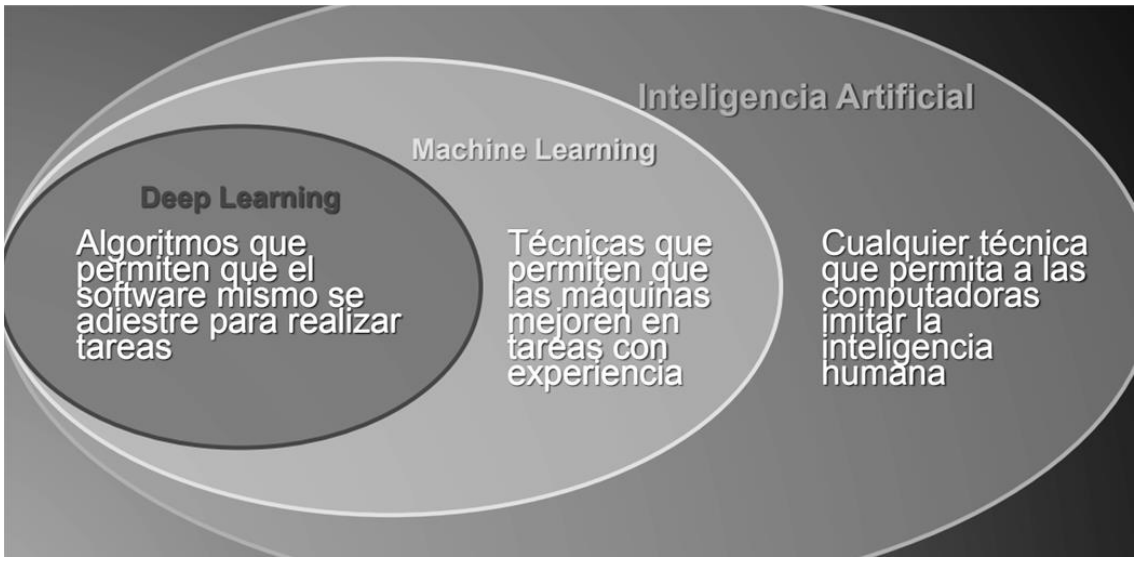

Figura 12: La Inteligencia Artificial
Fuente: Ruiz y Galvis (2018)

2.6.3. Machine Learning

El *Machine Learning* (Aprendizaje Automático), es un concepto relacionado con el *Data Mining*, aunque tiene sus diferencias en cuanto a origen, aplicación y objetivos. El *Machine Learning* tiene como objetivo desarrollar modelos computacionales capaces de inducir conocimiento a partir de experiencia previa (González *et al*., 2014). Es decir, se entrena los modelos para intentar predecir algún fenómeno particular (Educba, s.f.).

Técnicamente el *Machine Learning* hace parte de la inteligencia artificial (la cual busca que un programa determinado actúe, se adapte y razone de la manera más parecida a un ser humano), y se ocupa del aspecto de aprender de manera automática para tomar decisiones acertadas (o por lo menos cercanas a las acertadas).

Los métodos que se usan en *Machine Learning* se dividen en dos principales clases, supervisados y no supervisados, tal como se muestra en la Figura 13:

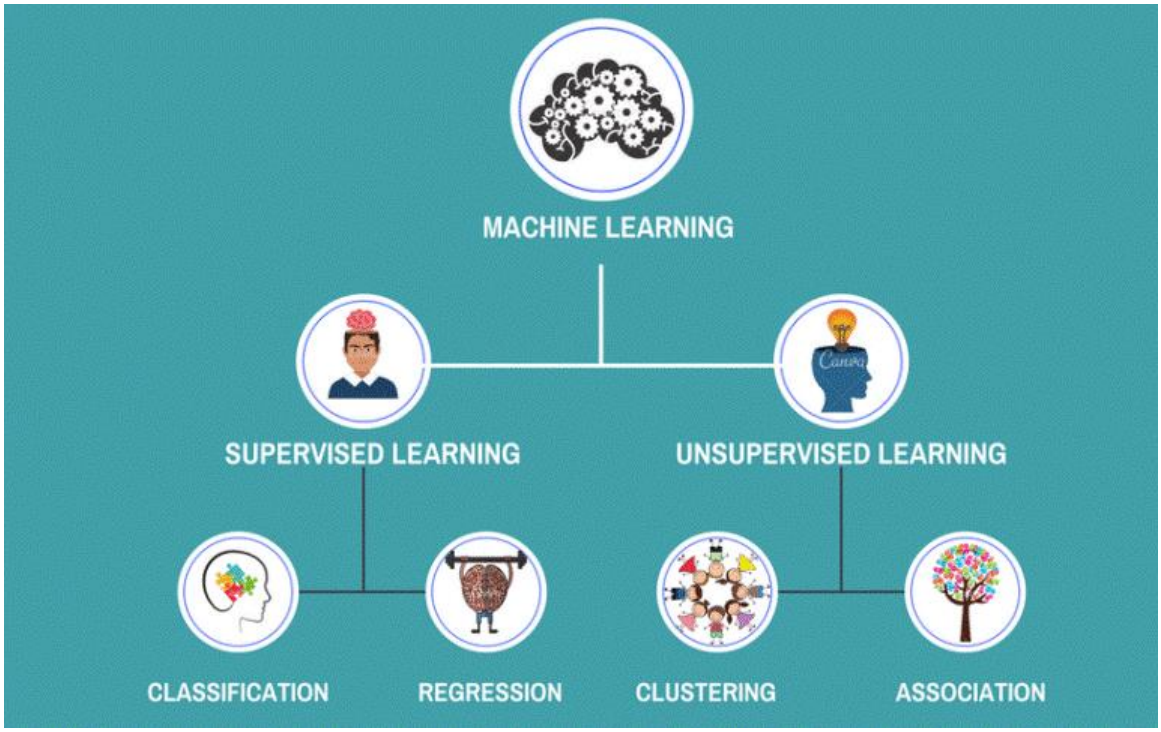

Figura 13: Paradigmas del *Machine Learning*
Fuente: Intellspot, citado por Almazán (2019)

2.6.4. Deep Learning

Se trata de una técnica de *Machine Learning* no supervisada, que usa una red neuronal artificial compuesta por varias capas. Eso le permite obtener un alto grado de éxito y permite ahorrar costos de supervisión humana en grandes conjuntos de datos. Sin embargo, puede consumir más recursos computacionales y tiempo (García, 2017).

Según el portal Bismart (s.f., ¶ 3), “Mientras que el *Machine Learning* trabaja con algoritmos de regresión o con árboles de decisión, el *Deep Learning* usa redes neuronales que funcionan de forma muy parecida a las conexiones neuronales biológicas de nuestro cerebro”. Las diferencias se pueden ver de manera más clara en la Figura 14:

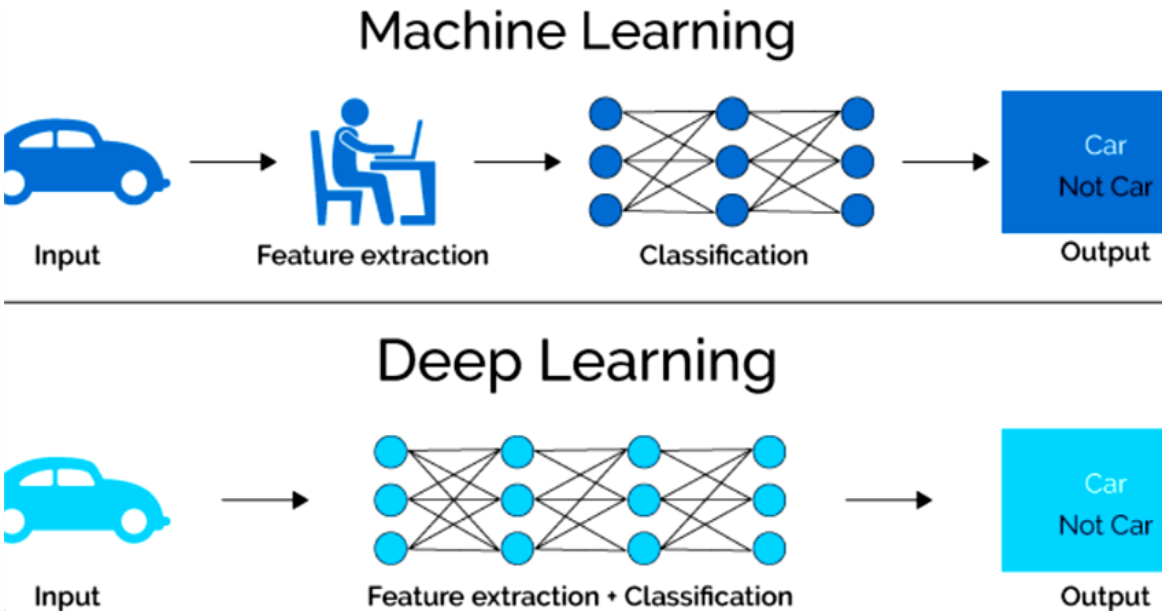

Figura 14: Diferencia entre *Machine Learning* y *Deep Learning*

Fuente: Ruiz y Galvis (2018)

En resumen, como lo expresa Cabana (s. f, ¶ 7), "el *Deep Learning* es una nueva rama del *Machine Learning*, que a su vez es una rama de la Inteligencia Artificial. El Big Data es un conjunto de tecnologías independientes que no tienen por qué estar relacionadas con la Inteligencia Artificial y que obedecen más al concepto de gestión de multitud de datos. Entonces, ser un científico de datos (*data scientist*), requiere tener un dominio medio-alto de inteligencia artificial, *machine learning*, *deep learning* y *big data*".

2.7. DATA SCIENCE Y SISTEMA DE INFORMACIÓN GEOGRÁFICA

La creciente disponibilidad de información espacial ha posibilitado la aplicación de las técnicas de Inteligencia Artificial y *Machine Learning* al campo de los Sistemas de Información de Geográfica (SIG). Al respecto Abarca et al. (2017) recogen experiencias de este tipo en temas de planificación urbana y territorial, así como también geografía humana y urbana que se han tratado con aprendizaje no supervisado, mapas auto organizados y construcción mediante aprendizaje

supervisado con árboles de decisión.

Como resultado de su investigación advierte que la integración completa entre la creación de Mapas Auto-organizados (SOM) y SIG todavía no se ha implementado de manera efectiva para que se pueda trabajar de manera directa y amigable en los principales programas SIG (Abarca et al., 2017).

A pesar de ello, de acuerdo a la recopilación de literatura y su propia experiencia durante su investigación, Abarca et al. (2017) concluyen que la metodología SOM tiene varias utilidades como lo es por ejemplo: realizar análisis exploratorio, comprender los patrones de distribución espacial, analizar complejos conjuntos de datos geográfico-demográficos; inferir consideraciones espaciales a partir de los grupos taxonométricos; codificar las clasificaciones resultantes en un SIG consiguiéndose hacerlos más accesibles y comprensibles; etiquetar la realidad geográfica sin tener que nombrar tales categorías, entre otras.

Abarca et al. (2017) también evaluaron la metodología basada en árboles de decisión a partir de una clasificación SOM y concluyeron que son útiles para: atribuir patrones de comportamiento que pueden ser muy complejos; predecir comportamientos de variables que presentan cierto coste o dificultad de evaluación, como son las variables demográficas o sociales, a partir de otras variables con menor complejidad y coste de evaluación, como las variables residenciales; identificar variables que se relacionan de forma significativa y su peso o tamaño del efecto en la realidad estudiada.

Actualmente algunos programas para Sistemas de Información Geográfica están incorporando nuevas herramientas que permiten aplicar los conceptos de la ciencia de datos y especialmente el *Machine Learning* al análisis espacial. Como el caso del software licenciado ArcGIS, que de acuerdo con Ruiz y Galvis (2018), incluye funcionalidades como las siguientes:

Clustering: Agrupación de observaciones basadas en similitudes de valores o

ubicaciones: agrupamiento multivariado, agrupamiento basado en densidad, segmentación de imágenes y análisis Hot Spot.

Clasificación: Decidir a qué categoría se debe asignar un objeto en función de los datos: Hallar ubicaciones adecuadas, identificar la mejor ruta existente entre ubicaciones y cálculos zonales.

Predicción: Usar lo conocido para estimar lo desconocido, modelos determinísticos, interpolación de áreas y mapas de probabilidad.

2.8. SMART CITIES E IDES

Las *Smart Cities* (Ciudades Inteligentes), “son las que usan las tecnologías de la información y las comunicaciones para hacer que su infraestructura crítica y de servicios públicos sean más interactivos y eficientes” (Bustillo y Rodríguez, 2015, p. 258).

Según Bustillo y Rodríguez (2015), el crecimiento de la disponibilidad de datos se da hoy en día gracias a la multitud de plataformas para la adquisición de datos como son: tecnología de multisensores, informática móvil, *cloud sourcing*, aplicaciones 3D y posicionamiento satelital; haciendo más fácil la posibilidad de descubrir fenómenos espaciales con alto grado de precisión geográfica y temática.

Para Almazán (2019), implementar una *Smart City* requiere de varias herramientas y conocimientos: políticos, económicos, administrativos, tecnológicos y técnicos. Y una de las herramientas más poderosas que puede tener es prescitamente la Inteligencia Artificial.

Las fuentes de información y su disponibilidad están creciendo de manera importante, y ahora el reto es almacenar y procesar esa información con la calidad suficiente. Para ese propósito, las IDE (o su concepto), pueden ser de gran utilidad, ya que los estándares permiten la interoperabilidad con distintos sistemas y se pueden poner a disposición de los diferentes usuarios. Además la calidad de la información se garantizar de mejor manera, igualmente se puede hacer

seguimiento de su origen y políticas de distribución (Bustillo y Rodríguez, 2015).

Muchas ciudades en el mundo ya se están guiando por las IDE como modelo o también como eje central para su transformación hacia el concepto de ciudades inteligentes, especialmente en Europa. Tal es el caso del proyecto “Sevilla ciudad inteligente”, que busca aprovechar las fuentes de datos espaciales oficiales más los datos generados por la interacción ciudadana, para mejorar su calidad de vida en sectores como el transporte, la vivienda y los servicios públicos (Velazco, Abuchar, Alzate; 2017).

Al respecto, el portal UrbanismoSevilla (s.f., ¶ 1) recoge lo siguiente: “la IDE de Sevilla tiene como misión aglutinar cuatro áreas relacionadas con la producción y gestión de los datos espaciales, su análisis y publicación: la producción de la Cartografía de Sevilla, la Infraestructura de Datos Espaciales del Ayuntamiento, el Sistema de Información Geográfica corporativo y la Delineación e Infografía de todo tipo de planes y proyectos. Las cuatro áreas participan de manera coordinada junto con el resto del Ayuntamiento, en el desarrollo de la "Plataforma Smart City de Datos Espaciales".

De manera tal que el tratamiento de grandes volúmenes de información, a pesar de los retos que implica, ya se está beneficiando de las IDE y sus estándares para lograr hacer un mejor proceso de los datos y entregarlos de manera rápida, fácil y con alta confiabilidad a sus usuarios y ciudadanía en general. En este sentido, el modelo de las IDE, con sus estándares y buenas prácticas en el manejo de la información geográfica, se convierte en un referente a tener en cuenta en las diferentes técnicas de la ciencia de datos y en especial aquellas que más se relacionan con los SIG, como lo es el concepto de ciudades inteligentes.

3. METODOLOGÍA

Para dar respuesta a los objetivos de esta investigación del presente proyecto se tuvo en cuenta la metodología de Demšar y Zupan (2013), quienes dividieron su metodología de trabajo en varias fases. De tal manera que una adaptación al presente estudio sería la siguiente:

1. Definición de la zona de estudio.
2. Caracterización de la información espacial asociada.
3. Análisis del documento resultante de la consulta GetCapabilities
4. Lectura del documento GetCapabilities mediante el script.
5. Elaboración de URL de descarga a partir del documento GetCapabilities mediante Python.
6. Descarga de la Información.

3.1. FLUJOGRAMA DE LA METODOLOGÍA

La *Figura 15*Figura 15 se representa el flujograma ejecutado:

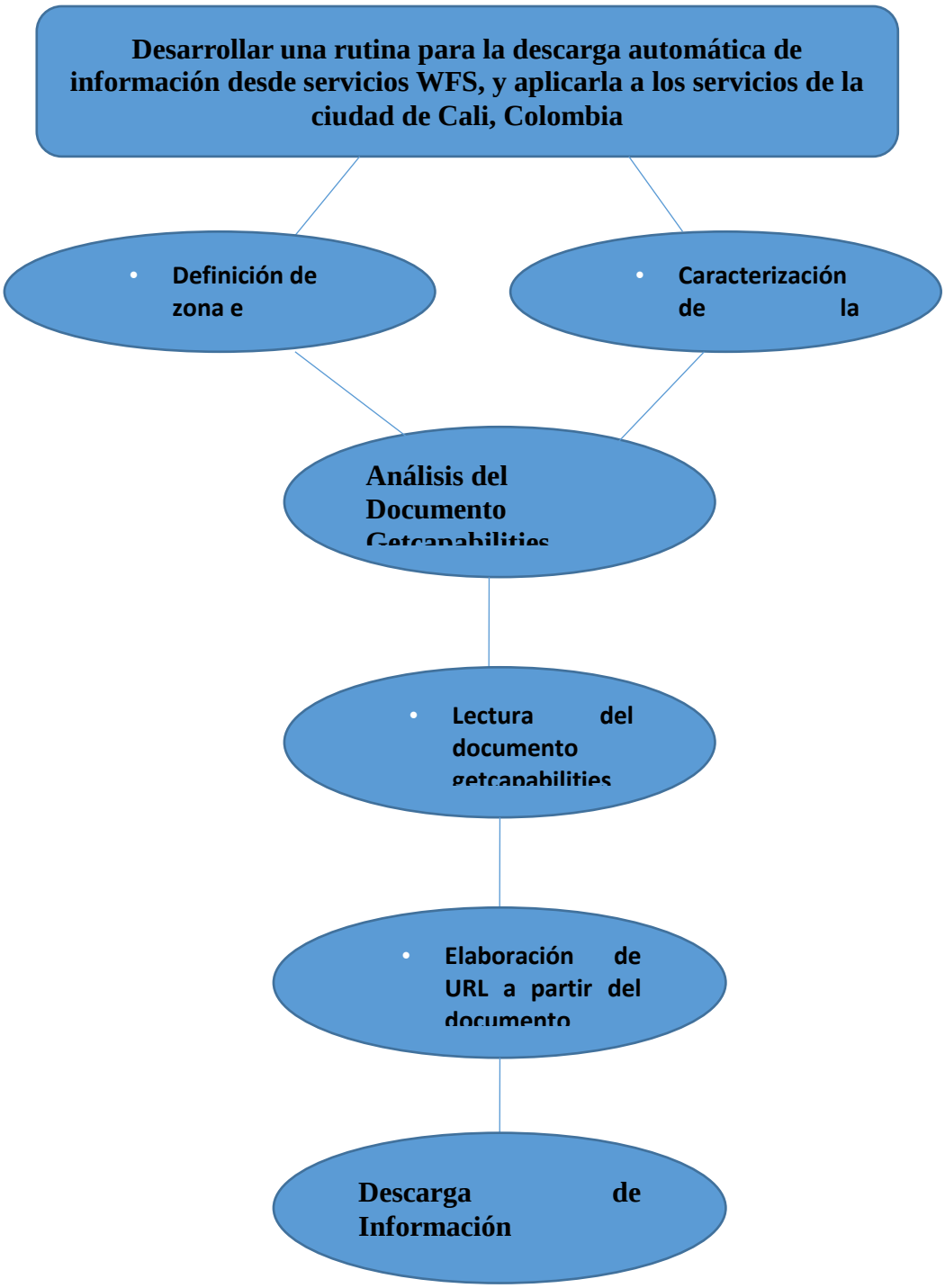

Figura 15: Diagrama del proceso metodológico de la presente investigación

3.2. DEFINICIÓN DE ZONA DE ESTUDIO

El área de estudio es el municipio de Santiago de Cali (conocido como Cali), Colombia; con coordenadas 3°27′00″N 76°32′00″O. Está situado en la

región suroccidental y junto al río Cauca (el segundo más importante de la nación), por lo que tiene una posición geográfica estratégica, ubicándose entre las ciudades principales de Colombia.

A continuación, la *Figura 16*Figura 16 ilustra el mapa de la zona de estudio:

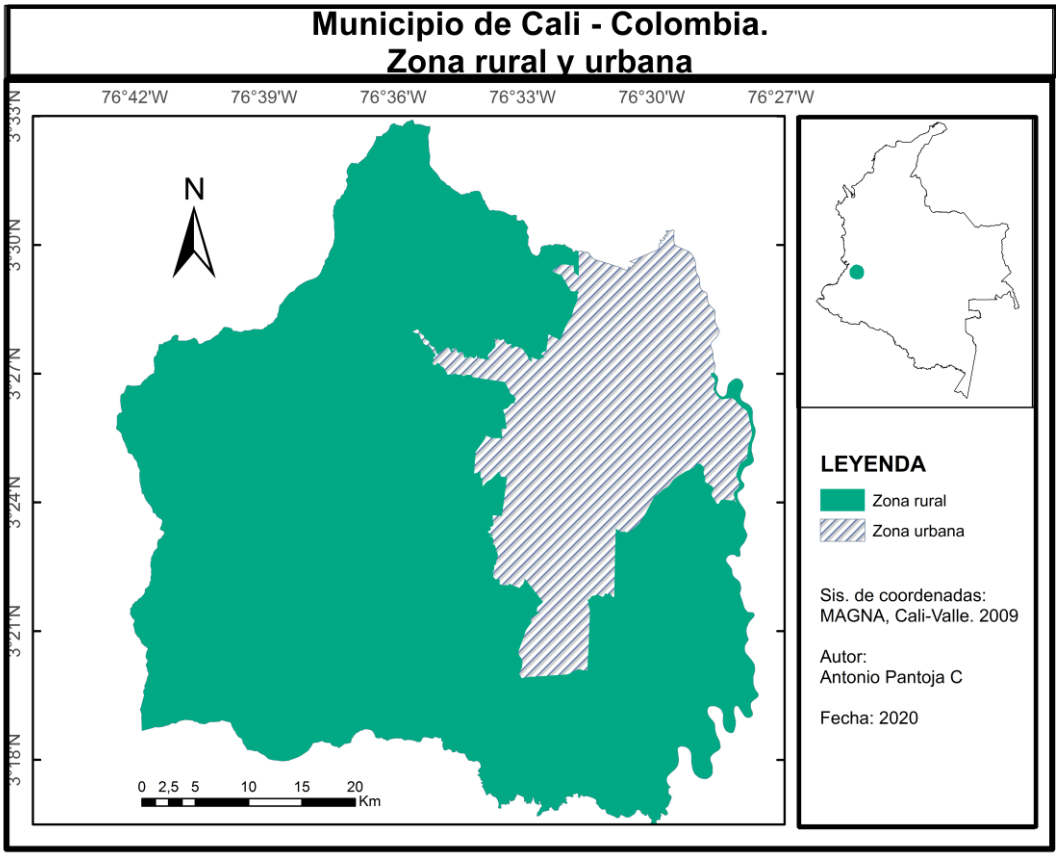

Figura 16: Zona de estudio: Cali, Colombia

El municipio de Cali es la capital del departamento del Valle de Cauca, que cuenta administrativamente con 42 municipios, y se encuentra a 484 Km. de Santafé de Bogotá, capital del país.

3.3. ORIGEN DE LA INFORMACIÓN GEOGRÁFICA

La información geográfica se obtuvo usando la petición *GetCapabilities* al servicio WFS de la IDE de Cali. En el contexto que se describe a continuación:

IDE Cali

Dado el ámbito suscrito al municipio de Cali, Colombia; se utilizó la Infraestructura de Datos Espaciales de Cali (IDESC), para acceder a la información geoespacial disponible, y realizar diferentes pruebas, como lo es conocer el nombre de las capas geográficas disponibles, y de esa manera construir una metodología de descarga automatizada. El URL de acceso inicial es: http://ws-idesc.cali.gov.co:8081/geoserver/wfs?

***Web Feature Service* (WFS)**

El servicio WFS permitió establecer la comunicación y conocer las capas de información disponibles de la IDE. Para realizar estas operaciones se utilizó la petición *GetCapabilities*, el cual genera un documento en el lenguaje GML, y es el estándar a través del que se transmiten la ordenes WFS. Así se pudo acceder a la información que describe los datos geoespaciales vectoriales contenidos en la IDESC.

Librerías de Python utilizadas

El submódulo *request*, del módulo *urllib*, permitió leer el contenido del documento (en formato XML) resultante de la petición *GetCapabilities*. Por otra parte, el módulo *urlretrieve* (usado como urllib.urlretrieve) permitió descargar cada capa, usando el URL completo (con todos los parámetros) e indicando el directorio local de descarga.

3.4. ANÁLISIS DEL DOCUMENTO *GETCAPABILITIES*

En el documento GML resultante de la petición *GetCapabilities* se puede ver el nombre de las capas que están disponibles para ser descargadas. Este servicio WFS, a diferencia del formato WMS, tiene la posibilidad de descargar las capas de información, si se conoce el nombre de la capa de interés y se construye el URL correcto para ello.

Entonces, se usó Python para dos propósitos, i) determinar los nombres de las capas y así conocer la cantidad de datos disponibles. Para ello se usó las librerías *Urlopen y Urllib*, que permiten leer textos a partir de un URL y el contenido de su página web. Y ii), construir mediante una variable temporal el URL completo de la capa, para descargarla mediante el uso de la librería *urllib*.

3.5. LECTURA DEL DOCUMENTO *GETCAPABILITIES* CON EL SCRIPT

El script Python creado permite leer cadenas de caracteres y mediante el uso de comodines (palabras o caracteres claves), separar bloques de texto. Así se obtuvieron los nombres de las capas, buscando la palabra clave *Name*, en el documento resultante de la petición *GetCapabilities*. Esos nombres son usados para completar el URL que permite la descarga.

3.6. URL A PARTIR DEL DOCUMENTO *GETCAPABILITIES*

Dado que el URL de descarga de una capa consiste en una serie de parámetros, se determinó que el nombre de la capa era uno de ellos y se trataba de un parámetro variable, que correspondía a cada una de las capas, de manera que, para construir el URL, se usó la concatenación de esos parámetros y el nombre de la capa, que se recuperó en un paso anterior (haciendo uso de la función *split*, y usando el comodín *name*). A ello se concatenó también el parámetro del formato de descarga, y fue así como se logró la descarga de cada una de las capas en un ciclo *for*.

De manera que Python permitió conocer la cantidad de capas disponibles, conocer sus nombres, y a la vez, descargarlas de manera rápida y eficiente.

3.7. DESCARGA DE INFORMACIÓN

Para obtener el URL completo de descarga se utilizaron funciones propias de

Python, y en este caso particular, el operador de suma (+), para incorporar los parámetros del servidor, el nombre de la capa y el formato del archivo a descargar.

Para el desarrollo de esta rutina se puso en práctica elementos teóricos- prácticos. La parte teórica supuso la revisión exhaustiva de los sustentos teóricos fundamentales de la IDE y el documento resultante de la petición *GetCapabilities*. Pudiéndose obtener una gama de información detallada que sustenta la arquitectura de la herramienta. Los elementos prácticos, a su vez, provienen de las experiencias profesionales relacionadas a la descarga de datos y tecnologías de la información geográfica.

3.8. ARQUITECTURA DEL PROCESO

La *Figura 17*Figura 17 ilustra la arquitectura ejecutada para el desarrollo de la rutina:

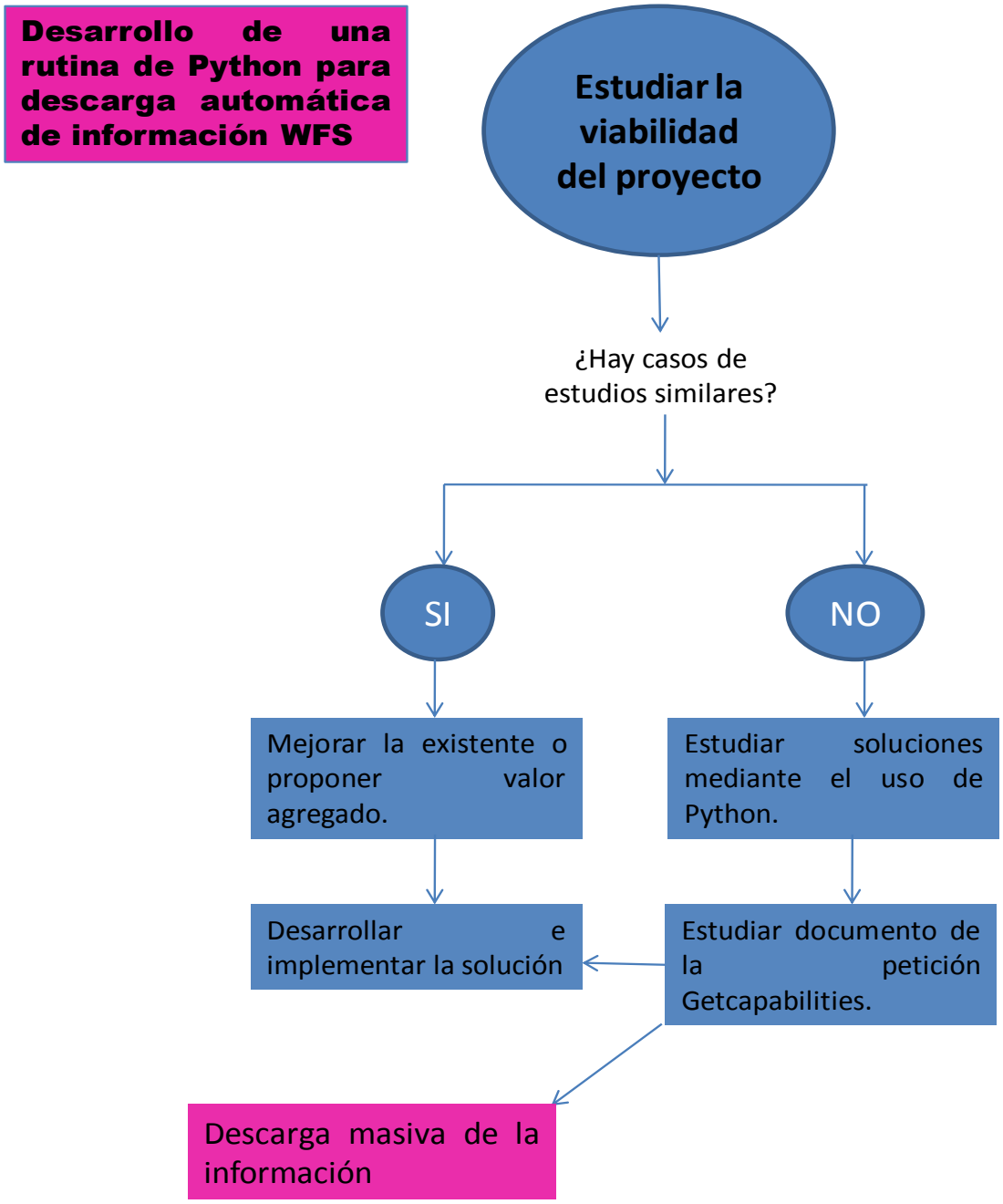

Figura 17: Arquitectura del proceso

4. RESULTADOS

A continuación se describen los resultados obtenidos en la aplicación y desarrollo de los objetivos de esta investigación:

4.1. DIPONIBILIDAD DE LA IG EN LA IDESC

Mediante la metodología utilizada accedió a la información contenida en el servicio WFS de la IDESC y se conoció la cantidad total de capas disponibles, que son 229 (al momento de la aplicación del proceso). Además, se pudo conocer también sus nombres de manera muy rápida (10 segundos).

Eso fue posible analizando los patrones generales del documento XML resultante de la petición *GetCapabilities*, para crear un script que realiza el proceso de coteo de las capas de manera automática. En la Figura 18 se muestra parte del código en Python (lado izquierdo), y los resultados (parte derecha en color negro).

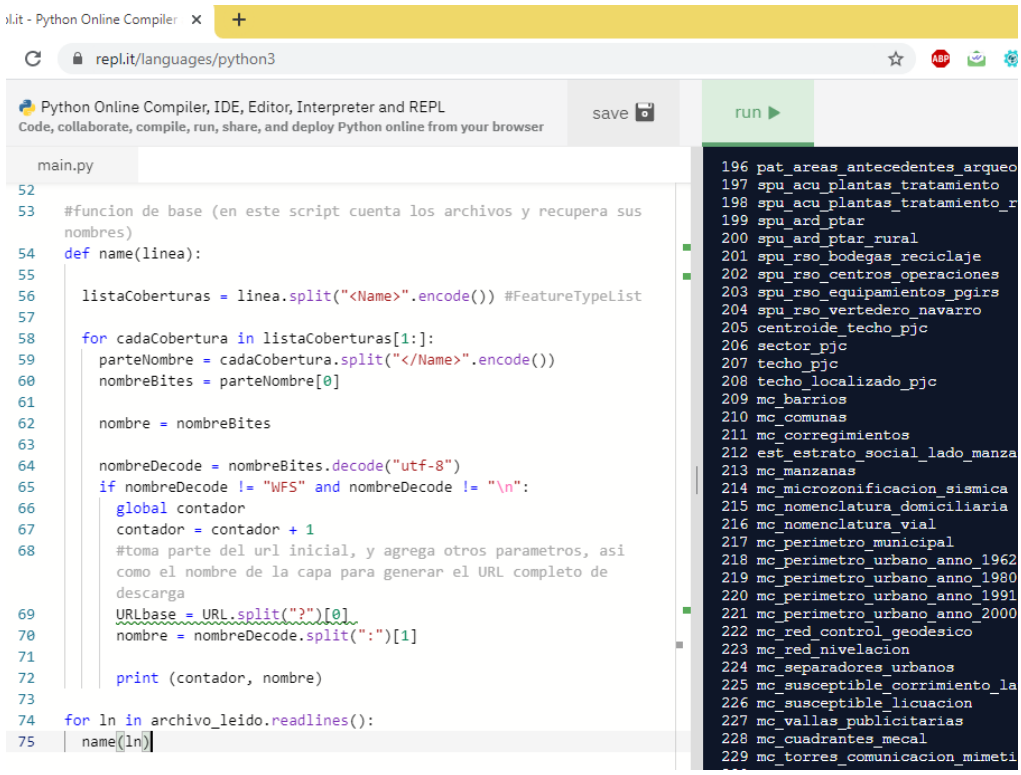

Figura 18: Conteo automático de capas WFS IDESC

4.2. METODO PARA EL ACCESO AUTOMÁTICO CON PYTHON

La característica clave del método de acceso radica en una función en particular que se creó para identificar los nombres de las capas. El código de la función creada en Python es el siguiente:

```
#funcion de base (en este script cuenta los archivos y recupera sus nombres)
def name(linea):

  listaCoberturas = linea.split("<Name>".encode()) #FeatureTypeList

  for cadaCobertura in listaCoberturas[1:]:
    parteNombre = cadaCobertura.split("</Name>".encode())
    nombreBites = parteNombre[0]

    nombre = nombreBites

    nombreDecode = nombreBites.decode("utf-8")
    if nombreDecode != "WFS" and nombreDecode != "\n":
      global contador
      contador = contador + 1
      #toma parte del url inicial, y agrega otros parametros, asi como el nombre de
la capa para generar el URL completo de descarga
      URLbase = URL.split("?")[0]
      nombre = nombreDecode.split(":")[1]

      print (contador, nombre)
```

Patrón general de la rutina de Python

El script completo para el proceso de lectura de la cantidad de capas y sus nombres se deja a continuación. Se puede hacer uso de él incluso en un editor de código Python en línea como lo es por ejemplo https://repl.it/languages/python3

```
#Contador nombres
#Verificar la cantidad de capas disponibles y sus nombres, en el WFS de la
IDESC
#Febrero 2020. AntonioPantojaC@gmail.com

# URL de la peticion GetCapabilities
URL =
r'http://idesc.cali.gov.co:8081/geoserver/wfs?&request=GetCapabilities&service=w
fs&version=1.0.0'
pathDescargas= r'D:\Descargas_IDESC1'


#importar librerias
import os,sys,os.path,re,urllib
from urllib.request import urlretrieve



#Librerias para leer URL, en Python 3
from urllib.request import urlopen


#borrar historia de la ventana CMD
os.system('cls')


#funcion para probar si el URL tiene todos los parametros requeridos
def tesTUrl(parametro,texto):
  PresenciaParametro=re.findall(parametro,texto)
  if len(PresenciaParametro) == 0:
    print ("\nEn el URL falta: ", parametro , "  o no esta bien escrito (verificar
mayusculas y minusculas\n\n")
    sys.exit()
```

```
testUrl("request=GetCapabilities",URL)
testUrl("request=",URL)
testUrl("request",URL)

testUrl("service=wfs",URL)
testUrl("service=",URL)
testUrl("service",URL)

testUrl("version=",URL)
testUrl("version",URL)



# Leyendo el URL , con Python 3
archivo_leido = urllib.request.urlopen(URL)


#desde el URL recuperar sun nombre y version del servidor
nombreURL =  URL.split('.')[1]
valorVersionWFS = URL.split('version=')[1][:5]


contador = 0

#funcion de base (en este script cuenta los archivos y recupera sus nombres)
def name(linea):

  listaCoberturas = linea.split("<Name>".encode()) #FeatureTypeList

  for cadaCobertura in listaCoberturas[1:]:
    parteNombre = cadaCobertura.split("</Name>".encode())
    nombreBites = parteNombre[0]
```

```
    nombre = nombreBites

    nombreDecode = nombreBites.decode("utf-8")
    if nombreDecode != "WFS" and nombreDecode != "\n":
      global contador
      contador = contador + 1
      #toma parte del url inicial, y agrega otros parametros, asi como el nombre de
la capa para generar el URL completo de descarga
      URLbase = URL.split("?")[0]
      nombre = nombreDecode.split(":")[1]

      print (contador, nombre)

for ln in archivo_leido.readlines():
  name(ln)

#Fin del codigo
```

La aplicación de la rutina y las variaciones a la misma a nivel de código permitieron también descargar las capas en formato comprimido .zip, para ser almacenadas de manera local, tal como se puede ver en la *Figura 19*Figura 19, que muestra los resultados al usuario tras la ejecución (parte superior de la figura), y las capas en el explorador de Windows (parte inferior de la figura).

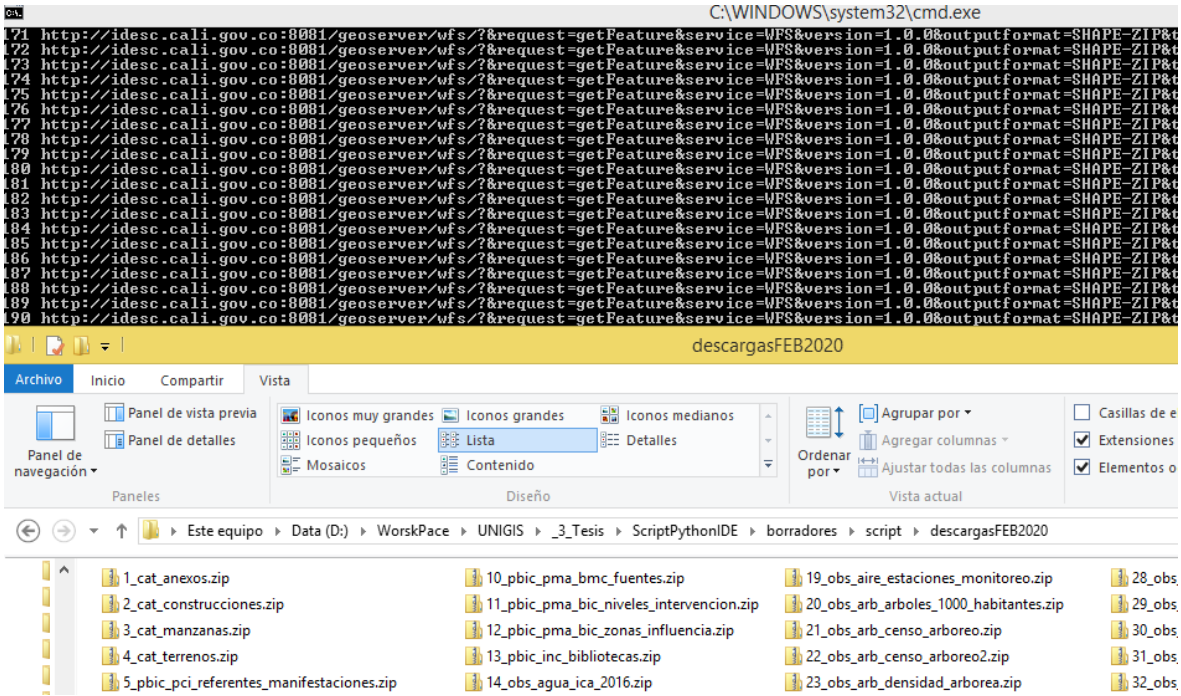

Figura 19: Capas descargadas desde la IDESC

5. DISCUSIÓN

En este capítulo se desarrolla el espacio de discusión bajo las siguientes aristas: disponibilidad de la información, recolección de nombres de las capas WFS, descarga de las capas WFS con QGIS, descarga de las capas WFS aplicando la rutina de Python y su funcionalidad.

Tomando en cuenta el estudio de Demšar y Zupan (2013), quienes indican que Python como lenguaje de programación es bastante simple y fácil para generar scripts, se pudo comprobar mediante la ejecución de la rutina que, con el uso de las librerías de Python, se accedió a la información contenida en el servicio WFS de la IDESC, permitiendo la descarga automática de ella, la cual puede ser dispuesta por el operador o usuario para su posterior uso, según sea su requerimiento.

En cuanto al proceso de verificar y evaluar la disponibilidad de la información geográfica en el portal de servicios de mapas WFS de la IDESC, la rutina de Python permitió conocer varios detalles para su descarga. En primer lugar, se pudo verificar el número de capas de forma automática. En el momento de la descarga de la información como tal con el proceso automatizado se pudo conocer también que algunas de estas capas tenían restricciones para su acceso y requieren de permisos adicionales para su descarga, por lo que no se pudo acceder a ellas. Hay que tener en cuenta también que la rutina desarrolla se limita a solamente conocer la cantidad de capas y sus nombres, pero no permite saber de manera anticipada si son libres para la descarga o no, sino hasta que se inicie como la solicitud al servidor y este la rechace.

Por otro lado, la rutina permite disminuir el tiempo del usuario frente al computador, por cuanto su funcionabilidad es automática, para que el usuario disponga de más tiempo para poder realizar otras tareas.

Para ejecutar la herramienta y descargar información o hacer adaptaciones es necesario que el usuario conozca, al menos a nivel básico, cómo ejecutar scripts de Python y algunos principios del lenguaje, ya que es mediante esta modalidad

como se va a ejecutar la rutina creada. Esto puede representar una desventaja, pero también puede ser una oportunidad para adquirir nuevos conocimientos relacionados Python, bajo la motivación de facilitar el trabajo de descarga de datos, mediante un proceso más práctico.

Además, la rutina aplicada permitió conocer los nombres de las capas existentes en la IDESC, con posibilidades de que se pueda aplicar esta herramienta a otras IDE, para lo cual es posible hacer algunas adaptaciones y, sobre todo, comprobar los resultados. Se puede pensar incluso en aplicar los mismos principios desarrollados a nivel de consulta de la información disponible del servicios WFS, a otros servicios de las IDE.

Es necesario aclarar que desde las IDE es posible descargar las capas WFS mediante programas de SIG, como lo es por ejemplo QGIS, pero de manera individual, es decir, capa por capa, pero con la rutina o script desarrollado el proceso de descarga se puede automatizar. Además, con esta rutina se pudo generar el listado de los nombres de las capas de manera automática y obtener los resultados de forma inmediata.

Para realizar una comparación de proceso de conteo de la información y su descarga se tomó como patrón realizarlo también desde el programa QGIS. Como resultado se presenta el cuadro comparativo de la tabla 2.

Tabla 2. Cuadro comparativo: descarga de datos con QGIS vs. rutina creada.

QGIS	Rutina de Python.
Se descargan las capas una a una.	Se descargan todas las capas existentes en el servicio WFS, automáticamente
El proceso de descarga es más lento, por incluir más pasos en el proceso	La descarga tiene un tiempo más corto.
No es sencillo ver todos los nombres de las capas disponibles.	Permite ver los nombres de las capas disponibles.
No permite conocer cuáles son las	Permite conocer de manera más rápida

capas restringidas a descarga, de manera inmediata.	cuáles son las capas restringidas a descarga.
Las capas se ven en la ventana del programa y se pueden descargar localmente	Las capas se guardan localmente de manera rápida y en su totalidad

Para saber cuántas capas están disponibles, usando QGIS, habría que contarlas una a una, siendo esto un inconveniente para el usuario ya que requiere mayor tiempo y verificación a detalle. Además, durante el proceso de descarga tendría que dedicarse exclusivamente a él, y aun así equivocarse. En cambio, con la rutina utilizada, el tiempo total de ejecución fue de una hora aproximadamente, donde el operador no requiere intervenir, es decir, que solamente invertiría unos pocos minutos en ejecutar la rutina para que ella realice el proceso de manera automática

Pero también se tomó en cuenta algunas desventajas de la aplicación de esta rutina o herramienta en comparación con el programa informático QGIS, tal como lo describe la tabla 3:

Tabla 3. Cuadro comparativo: desventajas de la rutina con respecto a QGIS

QGIS	Rutina de Python.
No se necesita conocer el documento GetCapabilities	Idealmente se debe conocer el documento GetCapabilities, su contenido o su interpretación
El usuario no necesita conocer la ejecución de scripts de Python.	El usuario necesita conocer la ejecución de scripts de Python para aplicar la rutina
Puede seleccionar qué capas descargar	Se descarga todas las capas, sin poder elegir cuáles sí y cuáles no
Las capas quedan almacenadas a nivel local listas para ser usadas	Las capas quedan almacenadas en formato comprimido (.zip), por lo que se las debe descomprimir para ser usadas

Tal como se menciona en la Tabla 3, una desventaja de la rutina creada es que los archivos descargados están en formato ZIP, por lo que se las debe descomprimir, y esto genera un almacenamiento doble de información. Lo cual puede ser un inconveniente si no se dispone de mucho espacio de almacenamiento local. En cambio, con QGIS, cuando se descarga la información se guarda una sola vez, y en formato listo para ser usando, no se tiene que descomprimir.

Otro aspecto a mejorar en una nueva versión de la herramienta desarrollada es que se pueda seleccionar las capas de interés para ser descargadas, ya que la rutina actual no lo permite.

Es necesario tener en cuenta que el proceso de descarga de la información automática puede tardar un tiempo variable dependiendo de la conexión de internet con la que se cuente. Para la cantidad de datos que se encontró en este caso, el tiempo fue de una hora. Sin embargo, no se hicieron pruebas con diferentes tipos de conexión, solamente con diferentes horarios, lo cual arrojó que el proceso puede tardar mucho tiempo quizá más del esperado dependiendo de la congestión de la red.

Además de analizar y discutir de forma crítica los resultados, se presentan a continuación las respuestas a las preguntas de investigación planteadas al inicio de este documento:

«PI - 1» ¿Cuál es la disponibilidad de información geográfica en el servicio WFS de la IDESC? Partiendo de la conceptualización de una IDE como una estructura que permite el aprovechamiento de la información geográfica de manera eficiente, se tiene que la IDE de Cali, cuenta es una plataforma robusta, que tiene disponible un gran volumen de datos espaciales. Como se demostró en esta investigación, los servicios WFS de la IDESC permiten la descarga de los datos en formato vectorial, con una geometría descrita por un conjunto de coordenadas. Representando esto una opción para el uso de datos de manera útil en el ámbito de los SIG. La IDESC, dentro de su estructura, cuenta con convenios, actas, decretos, políticas de geoservicios, entre otros, que brindan confianza a quienes accedan al este portal web. En esta investigación se pudo trabajar de manera segura y confiable con toda

la amplia gama de información que ofrece este geoportal.

«PI - 2» ¿Se puede aplicar un método para acceder y descargar de información de manera automática desde el servicio WFS de las IDESC? Sí es posible aplicar al menos un método para este propósito. En este caso se creó una rutina para el acceso y descarga automatizada de las capas WFS, con base en la lectura del documento resultante de la petición GetCapabilities y la elaboración del URL a partir del mismo, mediante Python y el uso de distintas librerías de este que permiten realizar descarga automática de la información. Todo este proceso derivó en la creación de la rutina para descarga de datos WFS.

«PI - 3» ¿Cuáles serían las ventajas de aplicar un método de consulta y descarga de información de manera automática desde el WFS de la IDESC? De acuerdo a las pruebas y resultados del proceso, aplicar el método desarrollado y ejecutar una rutina de Python para la descarga de la información del servicio WFS de la IDESC, representa una gran ventaja debido a que el tiempo de descarga es mucho menor, si se compara con el tiempo que se emplea usando programas SIG, como por ejemplo QGIS. Además de eso, un proceso automatizado permite evitar posibles errores en los que puede incurrir el usuario al intentar hacer el proceso de manera manual, como lo es omitir alguna capa de interés de manera errónea.

Por otro lado, la rutina aplicada permitió ver los nombres de las capas, así como también aquellas cuyo acceso está restringido, de manera mucho más rápida, lo cual puede acortar tiempos para la gestión de los permisos necesarios en caso de que las capas disponibles sean de interés también.

En esta investigación se comprobó que es posible aplicar una rutina Python para la descarga de información WFS desde la IDESC, capa a capa en archivo zip. Sin embargo, la herramienta creada, aunque se utilizó exitosamente en dicha IDE, al tratar de usarla en otras IDE, se evidenciaron problemas con los nombres de las capas debido a caracteres especiales, o bien los URL base sugeridos en las páginas web de las IDE no siempre tenían la misma cantidad de parámetros iniciales. Esto evidencia la necesidad de acordar e implementar estándares que también tengan en cuenta las necesidades de automatización que algunos usuarios

requieren. De lo contrario, se puede limitar el potencial de aprovechamiento de este tipo de servicios.

6. CONCLUSIONES

La metodología aplicada confirmó la hipótesis inicial: se puede aplicar una rutina basada en el documento resultante de la petición GetCapabilities en el servicio WFS, de la IDESC, para descargar los datos disponibles mediante un proceso automatizado.

La aplicación de una herramienta para el manejo de datos derivados de una IDE es de gran utilidad para programadores o usuarios que deseen realizar la automatización de algunos procesos, como el de descarga de capas WFS. El lenguaje de programación Python es una muy buena elección para este tipo de procesos que involucra el acceso a la información almacenada en repositorios remotos.

El estándar WFS de acceso a archivos vectoriales permite consultar y acceder de manera automatizada a este, para generar listados con los nombres de dichos archivos o también para descargarlos.

La operación *GetCapabilities* permite conocer la cantidad de capas disponibles en el servicio WFS, que al momento de la consulta fueron 230 en total, y 229 disponibles para su descarga.

Hoy en día muchos usuarios de la información geográfica usan herramientas que les permiten descargar datos de manera masiva, por esta razón es necesario que estas herramientas sean simples y eficientes, de modo que se puedan concentrar en los problemas de sus respectivos campos. En este marco, el lenguaje Python, en particular, está cobrando importancia adicional al ser uno de los más utilizados en técnicas de tratamiento de datos a gran escala, por lo tanto, se considera que rutina creada en la presente investigación puede tener un interés práctico para los usuarios de SIG, pero también para la academia y público en general, como una manera de motivar el estudio de este lenguaje.

7. REFERENCIAS

Abarca, F., Campos, F., y Reinoso, R. (2017). *Metodología de ayuda a la decisión mediante SIG e Inteligencia Artificial: aplicación en la caracterización demográfica de Andalucía a partir de su residencia*. Estoa, Revista de la Facultad de Arquitectura y Urbanismo de la Universidad de Cuenca, Vol 6, No. 11, pp. 33-51. Accedido el 02 de febrero de 2020 en: http://dx.doi.org/10.18537/est.v006.n011.a03

Almazán, L. (2019). *La IA como vehículo del Desarrollo de Smart Cities*. Netcity.mx. Accedido el 02 de febrero de 2020 en https://netcity.mx/inteligencia-artificial-smart-cities/

Bahit, E. (2018). *Introducción al lenguaje Python*. Accedido el 15 de julio del 2019 en https://python.eugeniabahit.com/static/pdf/IntroduccionAlLenguajePython-Ed2018.pdf

Bermejo, E., y Conti, L., (s.f.). *Infraestructura de datos espaciales (IDE): ¿Qué es y por qué surge?.* Geoinnova.org. Accedido el 15 de julio del 2019 en https://geoinnova.org/blog-territorio/infraestructura-de-datos-espaciales-ide-que-es-y-por-que-surge/

Bernabé, M., y López, C. (2012). *Fundamentos de Infraestructura de datos espaciales (IDE)*. Accedido el 15 de julio del 2019 en http://redgeomatica.rediris.es/Libro_Fundamento_IDE_con_pastas.pdf

BiSmart (s. f.). *¿Cuál es la diferencia entre el machine learning y el deep learning?.* BiSmart.com. Accedido el 15 de julio del 2019 en https://blog.bismart.com/es/diferencia-machine-learning-deep-learning

Bustillo, E., y Rodríguez, P. (2015). *Los Sistemas de Información Geográfica y las ciudades inteligentes.* POLÍGONOS. Revista de Geografía, No 27, pp. 257-270. Accedido el 15 de julio del 2019 en http://revpubli.unileon.es/ojs/index.php/poligonos/article/view/3283

Cabana, E. (s. f.). *Inteligencia Artificial, Machine Learning, Deep Learning y Big Data.* aprendeconeli.com. Accedido el 15 de julio del 2020 en https://aprendeconeli.com/inteligencia-artificial-machine-learning-deep-learning-y-big-data/

Calderón, C., Barbosa, E., y Cabezuelo, L. (2016). *Técnicas Big data: análisis de textos a gran escala para la investigación científica y periodística.* El profesional de la información, 25(4), 623-631.

Castro, A., Sifuentes, E., González, S., y Rascón, H. (2013). *Uso de Minería de Datos en el manejo de Información Geográfica. Información tecnológica*. Vol. 25, No. 5, pp. 95-102. Accedido el 15 de febrero del 2019 en https://scielo.conicyt.cl/pdf/infotec/v25n5/art14.pdf

Cedeño, B., Mondragón, K., Moraga, J. C., y Solano, M. A. (2017). *EJE 04 -03 Infraestructuras de datos espaciales: Propuesta de implementación en la Facultad de Ciencias de la Tierra y el Mar, Universidad Nacional, Costa Rica.* Memorias Y Boletines De La Universidad Del Azuay, Núm.XVI Conferencia de sistemas de información geográfica. Accedido el 15 de febrero del 2019 en http://201.159.222.81/index.php/memorias/article/view/62

Challenger, I., Díaz, Y., y Becerra, R. (2014). *El lenguaje de programación Python*. Ciencias Holguín. Centro de Información y Gestión Tecnológica de Santiago de Cuba Holguín, Vol. XX, No. 2, pp. 1-13. Accedido el 15 de marzo del 2019 en https://www.redalyc.org/pdf/1815/181531232001.pdf

**Clinic Cloud (s. f*.). ¿Qué es el data mining? La definición de la minería de datos*. ClinicCloud.com. Accedido el 15 de marzo del 2019 en https://clinic-cloud.com/blog/data-mining-que-es-definicion-mineria-de-datos/

Covantec (2014). *Programación orientada a objetos.* Covantec. Entrenamiento Python Básico, Lección 9. Accedido el 15 de marzo del 2019 en https://entrenamiento-python-basico.readthedocs.io/es/latest/leccion9/poo.html

Demšar, J., y Zupan, B. (2013). *Orange: Data mining fruitful and fun - A historical perspective*. Informática (Slovenia). Vol. 37, pp. 55-60. Accedido el 15 de marzo del 2019 en https://www.researchgate.net/publication/289842192_Orange_Data_mining_fruitful_and_fun_-_A_historical_perspective

Díaz, J. (2016). *Sistemas de Información Geográfica (SIG): Python, Geoprocesamiento en ArcGIS*. Accedido el 15 de julio del 2019 en http://geomaticaysig.blogspot.com/2016/01/python-geoprocesamiento-en-arcgis.html

Educba (s.f.). *Difference between Data Mining and Machine Learning*. Educba.com. Accedido el 15 de julio del 2019 en https://www.educba.com/data-mining-vs-machine-learning/

García, J. (2017). *Python como primer lenguaje de programación textual en la Enseñanza Secundaria*. Evsal revistas. Education in the Knowledge Society (EKS), Vol, *18, No. 2*, pp. 147-162. DOI: 10.14201/eks2017182147162.Accedido el 15 de julio del 2019 en https://revistas.usal.es/index.php/eks/article/view/eks2017182147162

González, S., Gómez, I., Pastrana, J., y Hernández, A. (2014). *Algoritmos de clasificación y redes neuronales en la observación automatizada de registros.* Servicio de Publicaciones de la Universidad de Murcia. Vol. 15, No. 1, pp. 31-40. Accedido el 07 de marzo del 2019 en http://scielo.isciii.es/scielo.php?script=sci_arttext&pid=S1578-84232015000100003

Gour, R. (2019). *8 Key Differences between Data Science and Data Mining.* Ideca.gov. Accedido el 07 de marzo del 2019 en https://www.ideca.gov.co/recursos/glosario/global-spatial-data-infrastructure

Hansen, F. (2006). *Las IDE y el Comité Permanente para la Infraestructura de Datos Geoespaciales de las Américas, CP IDEA***.** Revista Cartográfica, No. 82, pp. 25-41.

IBM, International Business Machines Corporation (2009). *Elemento GetCapabilities: IBM Informix Dynamic Server (IDS), versión 11.50*. Accedido el 15 de mayo del 2019 en https://www.ibm.com/support/knowledgecenter/es/SSGU8G_11.50.0/com.ibm.dbext.doc/ids_dbxt_322.htm

IDE Canarias, Infraestructura de Datos Espaciales de Canarias (2015). *El OGC y el estándar WMS.* Accedido el 30 de julio del 2019 en https://www.idecanarias.es/documentacion/ogc_wms

IDECA (s.f.). *Global Spatial Data Infrastructure.* Glosario Ideca.gov.co. Accedido el 07 de marzo del 2019 en https://www.ideca.gov.co/recursos/glosario/global-spatial-data-infrastructure

IDEE, Infraestructura de Datos Espaciales de España (2017). *Introducción a las*

IDE. Accedido el 30 de julio del 2019 en https://www.idee.es/resources/documentos/Introduccion_IDEE.pdf

IDEE, Infraestructura de Datos Espaciales de España (2018). *Servicio de descarga (WFS) Versión 2.0.* Accedido el 30 de julio del 2019 en https://www.idee.es/resources/documentos/RD_wfs_v2_0.pdf

IDESC, Infraestructura de Datos Espaciales de Santiago de Cali (2016). *Guía de manejo geovisor IDESC*. Accedido el 30 de julio del 2019 en http://idesc.cali.gov.co/download/guias/guia_manejo_geovisor_idesc.pdf

IDESC, Infraestructura de Datos Espaciales de Santiago de Cali (2009). *Geoportal IDESC*. Accedido el 30 de julio del 2019 en https://www.cali.gov.co/planeacion/publicaciones/3560/idesc/

Iniesto, M., Núñez, A., González, J., Ariza, F., Ureña, M., Rodríguez, A., Abad, P., Rodríguez, J., Álvarez, M., Pérez, C., Bastarrika, A., Rodríguez, A., Torre, L., Manso, M., Rivas, D., Píriz, G., Coll, E., y Martínez, J. (2014). *Introducción a las Infraestructuras de Datos Espaciales*. DOI: 10.7419/162.12.2014

Itelligent (2018). *Deep Learning, Machine Learning e Inteligencia Artificial, ¿en qué se diferencian?* Intelligent.es. Accedido el 30 de julio del 2019 en: https://itelligent.es/es/diferencias-entre-deep-learning-machine-learning-inteligencia-artificial/

Jensen, K. (2012). *CRISP-DM Process Diagram.* CommonsWikimedia.org. Accedido el 15 de julio del 2019 en https://commons.wikimedia.org/w/index.php?curid=24930610

Junta de Andalucía (s.f). *Web Feature Service (WFS)*. Marco de Desarrollo de la junta de Andalucía. Accedido el 15 de julio del 2019. http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/32

Kulin, M., Kazaz, T., Moerman I., y Poorter, E. (2020). *A survey on Machine Learning-based Performance Improvement of Wireless Networks: PHY, MAC and Network layer*. Cornell University. Computer Science, Machine Learning. Accedido el 15 de julio del 2019 en ArXiv, abs/2001.04561.

Lages, G. (2018). *¿Qué es Data Science y cómo utilizar el análisis de datos para ampliar tu negocio?* HotmartBlog.com. Accedido el 15 de julio del 2019 en https://blog.hotmart.com/es/que-es-data-science/

Lemaître, G, Noguera, F., y Áridas, K (2016). *Imbalanced-learn: a python toolbox to tackle the curse of imbalanced datasets in machine learning*. The Journal of Machine Learning Research. Vol. 8, No. 1. ACMDigitalLibrary.org. Accedido el 2 de mayo del 2019 en https://dl.acm.org/doi/abs/10.5555/3122009.3122026

Mascarell, P. (2014). *Geoportal WEB IDE (Infraestructura de Datos Espaciales)*. [Tesis de pregrado], Universidad Politécnica de Valencia. Escola Tècnica Superior d'Enginyeria Informàtica. Accedido el 15 de mayo del 2019 en https://riunet.upv.es/bitstream/handle/10251/48237/MASCARELL-Geoportal%20WEB%20IDE%20%28Infraestructura%20de%20Datos%20Espaciales%29.pdf?sequence=3

Matos, G., Chalmeta, R., y Coltell, O. (2006). *Metodología para la Extracción del Conocimiento Empresarial a partir de los Datos*. Información tecnológica, Vol.17, No 2, pp. 81-88. Accedido el 15 de mayo del 2019 en https://dx.doi.org/10.4067/S0718-07642006000200011

Mesa, C. (2014). *La interoperabilidad como parte del desarrollo del Gobierno Electrónico en el Perú* [tesis de maestría]. Pontificia Universidad Católica del Perú. Accedido el 15 de mayo del 2019 en http://tesis.pucp.edu.pe/repositorio/bitstream/handle/20.500.12404/6721/MESA_TORRE_CRISTIAN_INTEROPERABILIDAD.pdf?sequence=1&isAllowed=y

Montes, C. (2008). *Los sistemas de información geográfica como medio didáctico en la enseñanza de la geografía* [tesis de maestría] Universidad de Antioquia - Facultad de Educación . Accedido el 15 de julio del 2019 en http://bibliotecadigital.udea.edu.co/bitstream/10495/7474/1/CarolinaMontes_2008_informaciongeografica.pdf

Olaya, V. (2014*). Sistemas de Información Geográfica*. Accedido el 15 de mayo del 2019 en https://volaya.github.io/libro-sig/

OSGeo Live (2011). *Estándares del Open GeoSPatial Consortium*. Osgeolive.org**.** Accedido el 15 de julio del 2019 en https://live.osgeo.org/archive/10.5/es/standards/standards.html

Pascual, P. (2017). *¿Cuáles son las diferencias entre Big Data y Data Science?* Piperlab.es. Accedido el 15 de julio del 2019 en https://piperlab.es/2017/12/05/diferencias-entre-big-data-data-science/

Ramos, J. (2016). *Diseño y desarrollo de un servicio Big Data en la nube para búsqueda, compartición de ficheros y data mining* [tesis de pregrado]. Universidad Complutense de Madrid (UCM)], Facultad de Informática. Accedido el 15 de julio del 2019 en Repositorio institucional: https://eprints.ucm.es/38504/1/MemoriaTFGJuanRamosDiaz.pdf

RedIRIS (2001). *Recetario para Infraestructuras de Datos Espaciales.* Redgeomatica.rediris.com. Accedido el 12 de abril del 2019 en http://redgeomatica.rediris.es/metadatos/publica/recetario/html/capitulo04_1.html

Ruiz, J., y Galvis, G. (2018). *GeoAI, Machine Learning y ArcGIS: Todo lo que necesita saber*. XX Conferencia Colombiana de Usuarios ESRI 2018. Bogotá, Colombia. Accedido el 10 de abril del 2019 en https://community.esri.com/docs/DOC-12440-geoai-machine-learning-y-arcgis-todo-lo-que-necesita-saber

Portolés, D., y Martínez, R. (2005). *La gestión de usuarios en una infraestructura de datos espaciales*. Instituto Nacional Geográfico JIDEE. Accedido el 10 de abril del 2019 en https://www.idee.es/resources/presentaciones/JIDEE05/sesion_07_04.pdf

Santamaria, W. (2006). *Técnicas de Minería de Datos Aplicadas en la Detección de Fraude: Estado del Arte*.Researchgate.net. Accedido el 12 de febrero del 2019 en https://www.researchgate.net/publication/240724702_Tecnicas_de_Mineria_de_Datos_Aplicadas_en_la_Deteccion_de_FraudeEstado_del_Arte

SGC, Servicio Geológico Colombiano (2017). *IDE geocientífica*. Accedido el 12 de febrero del 2019 https://www2.sgc.gov.co/sgc/ide-geocientifica/Paginas/ide.aspx

Snijders, C., Matzat, U., y Reips, U. (2012). *"Big Data": Big Gaps of Knowledge in the Field of Internet Science*. *International Journal of Internet Science*. Vol. 7, pp. 1-5. Accedido el 12 de febrero del 2019 en

https://www.researchgate.net/publication/231021899_Big_Data_Big_Gaps_of_Knowledge_in_the_Field_of_Internet_Science [

SNIT, Infraestructura nacional de Datos Espaciales de Costa Rica. (s.f.). *Acerca del SNIT*. Accedido el 12 de febrero del 2019 en https://www.snitcr.go.cr/about

Urbanismosevilla (s.f.). *Infraestructura Datos Espaciales (IDE): Presentación*. Accedido el 12 de febrero del 2020 en https://www.urbanismosevilla.org/areas/sostenibilidad-innovacion/ide

Vallejo, H., Guevara, E., y Medina, S. (2017). *Minería de Datos*. RECIMUNDO: *Revista Científica de la Investigación y el Conocimiento*. Vol. 2, No. Extra 1, 2018, pp. 339-349. Accedido el 12 de febrero del 2019 en https://dialnet.unirioja.es/servlet/articulo?codigo=6732870

Velazco, S., Abuchar, A., y Alzate, G. (2017*). Las infraestructuras de datos espaciales como apoyo al desarrollo de la ciudad inteligente*. *Redes de Ingeniería*. Edición especial 2016. Vol.74, No 10. Accedido el 12 de febrero del 2019 en http://hdl.handle.net/11349/20187