INSTITUTO TECNOLÓGICO SUPERIOR DE XALAPA

TÍTULO DEL PROYECTO
Caracterización de recursos computacionales para la fase de preprocesamiento de minería de textos

Opción de Titulación:
I Tesis profesional

Que como requisito parcial para la obtención de grado de

Maestría en Sistemas Computacionales

Presenta:

Yessenia Díaz Álvarez

No. De Control:
217O00013

Director
Dra. Virginia Lagunes Barradas

Co- Director
Dr. Miguel Ángel Hidalgo Reyes

Xalapa-Enríquez, Veracruz, octubre del 2023

DEDICATORIA

Primeramente, a Dios,

ya que gracias a Él puedo gozar de vida y salud, gracias por brindarme sabiduría, así como la compañía de personas que me han apoyado en todo momento a aprender de mis errores y a lograr lo que me propongo.

A mis padres, María Catalina y Jerónimo,

quienes son el pilar fundamental de mi vida, incluyendo la parte académica; gracias por estar en los días más difíciles, por haberme forjado como la persona que soy, por motivarme a alcanzar mis metas, por su incondicional apoyo a través del tiempo y porque muchos de mis logros se los debo a ustedes.

A mi tío Ángel Miguel,

quien siempre me ha apoyado en mis decisiones, quien ha sido un segundo padre para mí y un ejemplo a seguir; gracias por la constante motivación y apoyo brindado, fundamentales para conseguir mis metas.

A mis tíos Joaquín y Olegario,

quienes me han apoyado a salir adelante a pesar de cualquier obstáculo, gracias por estar siempre para mí, por aportar cosas buenas y estar al pendiente de mi vida.

A mi hermano Jesús Adrián,

por hacerme sentir mejor con su compañía, sus enseñanzas y su tiempo.

Este paso de mi carrera profesional, es dedicado a ellos, porque mis logros son reflejo de su dedicación.

AGRADECIMIENTOS

Al Consejo Nacional de Humanidades, Ciencia y Tecnología (CONAHCYT), por la beca otorgada dentro del proyecto “Caracterización de recursos computacionales para la fase de preprocesamiento de minería de textos” para atender problemas nacionales de ámbito ambiental.

A mi asesora Dra. Virginia Lagunes Barradas, quien me enseñó a desempeñar trabajos de calidad y valorar el estudio, superarme cada día, quien me ha ofrecido su conocimiento, me ha orientado, apoyado y corregido con un interés y entrega que ha sobrepasado en mucho mis expectativas.

Al Dr. Miguel Ángel Hidalgo Reyes por sus atenciones, así como por permitirme participar en el proyecto, por su guía y observaciones en este trabajo y por sus consejos para la vida profesional.

A la Mtra. María Angélica Cerdan y Mtra. Irma Angélica García González por ser parte de mi formación profesional, por su apoyo a lo largo de este tiempo y por su interés en formar estudiantes con valores y compromiso en cualquier circunstancia profesional.

A las investigadoras Dra. Obdulia Pichardo Lagunas y Dra. Bella Martínez Seis del Instituto Politécnico Nacional, así como a la Mtra. María de Lourdes Velasco Vázquez y Dra. Julia Aurora Montano Rivas de la Universidad Veracruzana, por darme la oportunidad de complementar mi desarrollo profesional, con su excelente asesoría en sus respectivas disciplinas y áreas de especialidad.

A mis maestros y compañeros, que me han brindado su apoyo incondicional, por los momentos compartidos, quienes estos años estuvieron para mí, logrando que esta meta se haga realidad.

RESUMEN

La minería de textos es un proceso encargado de extraer información proveniente de diversas fuentes de datos escritos. Sin embargo, previo a ello, deben llevarse a cabo diversas tareas de preprocesamiento, las cuales consumen recursos computacionales tales como tiempo y memoria. Esta tesis se enfoca, por un lado, en la evaluación en cuanto a tiempo y memoria, de dos de las técnicas utilizadas durante el preprocesamiento de textos aplicado a un conjunto de leyes ambientales mexicanas.

El realizar este tipo de procesamiento se debe a la necesidad de transformar texto no estructurado a un formato estructurado para identificar patrones significativos y tratar de descubrir nueva información. Además, las leyes ambientales se caracterizan por estar conformadas por un gran número de documentos legislativos existentes como leyes, programas y reglamentos, pudiendo surgir contradicciones derivadas de modificaciones por reformas y decretos. Por otro lado, se seleccionaron determinadas tareas de las metodologías Proceso Estándar entre Industrias para la Minería de Datos (*Cross Industry Standard Process for Data Mining, CRISP-DM)*, Proceso de Ciencia de Datos en Equipo *(Team Data Science Process, TDSP)* y la guía experimental de McGeoch, para conformar una metodología híbrida que permita establecer las pautas para la fase de preparación de datos en las tareas para la selección, limpieza, transformación y formateo. Dichas tareas, se ajustan a un diseño experimental en el cual se definieron las variables respuesta, tiempo (milisegundos) y memoria (bytes), los dos niveles de un solo factor, el tipo de biblioteca y como unidad de estudio, los nueve documentos referentes a leyes ambientales mexicanas. Finalmente, se comparó el tiempo y memoria de las bibliotecas PyPDF2 y Pdfplumber mediante la prueba de rangos con signos de Wilcoxon, lo cual dio como resultado diferencias en los tiempos de procesamiento de las leyes ($p<0.01$), con un valor medio de 5.43 ms

para la biblioteca PyPDF2 y de 58.26 ms para la biblioteca Pdfplumber. Con relación a la memoria utilizada no se presentaron diferencias.

Entre los principales hallazgos, los resultados mostraron que no existe relación entre el tamaño del documento y el consumo del tiempo. Además, en ambas bibliotecas se observó que sin importar el tamaño del documento el consumo de la memoria varía entre 60 bytes y 95 bytes. En particular la biblioteca Pdfplumber consume mayor hace mejor preprocesamiento de estos documentos

Xalapa, Veracruz, 20/julio/2023
Oficio N° OF/ITSX/SN

ASUNTO: Aceptación de Tesis de MSCO

MTRA. KORINA GONZÁLEZ CAMACHO
SUBDIRECCIÓN DE POSGRADO E INVESTIGACIÓN
PRESENTE

Los que suscriben, miembros del jurado, han realizado la revisión de la Tesis del C.:

Yessenia Díaz Álvarez

la cual lleva el título de:

Caracterización de recursos computacionales para la fase de preprocesamiento de minería de textos

Y concluyen que después de intercambiar opiniones, los miembros del jurado manifestaron ***APROBAR LA TESIS***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

A T E N T A M E N T E

PRESIDENTE: **Dra. Virginia Lagunes Barradas** FIRMA

SECRETARIO: **Dr. Miguel Ángel Hidalgo Reyes** FIRMA

VOCAL : **M.C. María Angélica Cerdán** FIRMA

VOCAL SUP. : **M.C. Irma Angélica García González** FIRMA

EGRESADA DE LA **MAESTRÍA EN SISTEMAS COMPUTACIONALES**

OPCIÓN: **I Tesis**

ccp. Archivo

CONTENIDO

Índice de Tablas

Índice de Figuras

CAPÍTULO I. GENERALIDADES DEL PROYECTO

1.1 ANTECEDENTES

La ciencia de datos en ocasiones conocida como descubrimiento de conocimiento, aprendizaje automático, análisis predictivo o incluso minería de datos, se refiere a un conjunto de técnicas utilizadas para extraer valor de los datos y esto resulta de utilidad para muchas organizaciones.

Para efectos de claridad, es importante distinguir entre los términos descubrimiento de conocimiento y minería de datos. El primero, conocido por su nombre en inglés como *Knowledge Discovery from Databases (KDD)*, se refiere al "proceso general de descubrir conocimientos útiles a partir de datos" *(Fayyad et al., 1996)*.

Por su parte, la minería de datos es una etapa del proceso *KDD* que consiste en "aplicar algoritmos de análisis de descubrimiento de datos que producen una determinada enumeración de patrones (o modelos) sobre los datos" (Fayyad et al., 1996), la cual es sólo una fase del proceso de *KDD* (Véase Figura 1), al igual que la selección, preprocesamiento, transformación e interpretación adecuada de los resultados del proceso *KDD*.

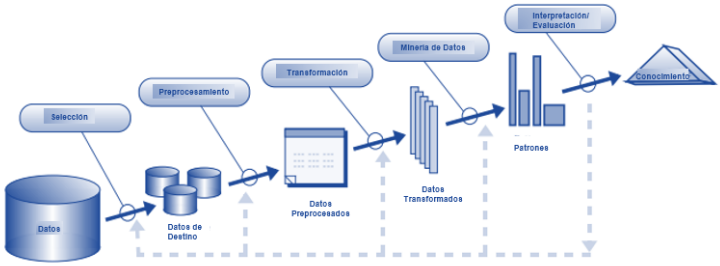


Figura 1. Pasos que componen el proceso de KDD (Fayyad et al., 1996).

Por lo general, la minería de datos se enfoca a la extracción de datos a partir de bases de datos o repositorios, sin embargo, y generalmente la información contenida se encuentra estructurada o apegada el modelo relacional de las bases de datos.

El acelerado crecimiento de la *WWW* (*World Wide Web, de sus siglas en inglés*), el de las transacciones financieras, las interacciones del usuario, así como la aparición de paradigmas como el *IoT* (*Internet of Things, de sus siglas en inglés*), ha hecho cada vez más necesario para dichas organizaciones utilizar herramientas automatizadas que le permitan encontrar, extraer, filtrar y evaluar la información disponible.

La minería de datos, acorde a la naturaleza de la información, se clasifica en categorías como la minería web, la minería de procesos, la minería multimedia y la minería de textos[1]. En este último caso se define como “el proceso de extraer el conocimiento implícito de datos textuales” (Barrera, 2014).

Adicionalmente es importante mencionar que “la minería de textos surge a principios de los años 80 cuando los textos empezaban a necesitar una gran cantidad de esfuerzo humano, para analizarlos. Entre los formatos de documentos utilizados se encuentran archivos doc/docx, archivos PDF y archivos HTML. El reto de esta categoría reside en analizar y modelar los contenidos de texto de lenguaje natural no estructurado. Por tal razón, se dice que la minería de textos es de facto una tecnología integrada de Procesamiento del Lenguaje Natural (*PLN*), clasificación de patrones y aprendizaje automático[2]” (Zong et al., 2021).

---

[1] *Text Mining, de sus siglas en inglés.*
[2] *Machine Learning, de sus siglas en inglés.*

El rápido crecimiento de los datos textuales en línea requiere del uso de diversas técnicas de minería de textos, tales como “clasificación y agrupación de textos, modelado de temas, análisis del sentimiento del texto y minería de opinión, detección y seguimiento de temas y extracción de información a partir de textos no estructurados y semiestructurados en lenguaje natural, entre otros” (Zong et al., 2021).

Sin embargo, para aplicar dichas técnicas, es necesario llevar a cabo ciertas tareas de preprocesamiento que tratan de transformar datos crudos, en datos que tengan formato y que, por lo tanto, sean más fáciles de utilizar. Algunas de estas tareas son descritas en la Tabla 1.

Tabla 1. Tareas de preprocesamiento (Zong et al., 2021).

Tarea	Descripción
Tokenización	Se refiere al proceso de segmentar un texto dado en unidades léxicas.
Remover palabras vacías	Las palabras vacías se refieren principalmente a palabras funcionales, incluidas palabras auxiliares, preposiciones, conjunciones, palabras modales y otras palabras de alta frecuencia que aparecen en diversos documentos con poca información textual.
Normas de formas verbales	El proceso de normalización de la forma de las palabras incluye la *lematización (stemming, en inglés)* es una técnica en la recuperación de datos en los sistemas de información (RDSI) y sirve para reducir variantes morfológicas para mejorar la habilidad de los motores de búsqueda.

Por otro lado, existen diversas herramientas que soportan entre sus funciones la minería de textos. Éstas pueden ser aplicaciones informáticas destinadas a dicho fin, o bien lenguajes de programación que incluyen entre sus bibliotecas funcionalidades para el análisis de textos.

Dentro de la categoría de lenguajes de programación, “*Python* es uno de los lenguajes enfocados a la minería de textos, albergando miles de módulos. Tanto la biblioteca estándar, como los módulos aportados por la comunidad, permiten posibilidades de análisis y además está desarrollado bajo una licencia de código abierto” (Python, 2021).

Un aspecto fundamental de este trabajo es el monitoreo de recursos computacionales y, al respecto su monitorización se define como “el proceso que extrae y almacena información acerca del estado de los recursos de un entorno computacional. Para ello, se definen un conjunto de métricas que miden y permiten conocer, en tiempo real, aquellos recursos que más interese tener controlados, dependiendo de la funcionalidad del sistema” (Palacio, 2015).

Además, la monitorización de recursos computacionales permite obtener información en tiempo y memoria al realizar su seguimiento y evaluación, para así facilitar la toma de decisiones.

Tras abordar de manera resumida los temas teóricos mencionados, existen dominios específicos donde se aplica todo lo anterior, y uno de ellos es la legislación, particularmente la política ambiental.

En este contexto, el proyecto Integralidad Gamma (Cuántico, 2020) que como lo mencionan (Pichardo et al., 2020) ha incursionado en el análisis de la legislación ambiental y destacan “que los datos, por sí solos, no proporcionan la información requerida para la adecuada toma de decisiones”. El equipo de Ciencia de Datos, de la Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas del Instituto Politécnico Nacional (UPIITA-IPN) en vinculación con el Instituto de Ecología A.C (INECOL), realizó un análisis de documentos legislativos en materia ambiental.

El propósito consiste en identificar automáticamente el nivel de coherencia de la jurisprudencia nacional implementando técnicas de minería automatizada de textos y preprocesamiento de lenguaje natural. Para abordar este proyecto se atendieron varios retos; por ejemplo, el primero fue la representación de la información, ya que no todas las leyes, reglamentos, normas, planes, programas, estrategias, acuerdos y lineamientos siguen la misma estructura. Lo anterior obliga a que los documentos sean procesados siguiendo un estándar de representación legible por computadora.

En la estructura a evaluar para la marcación de coherencias legislativas, también se debe analizar la relación jerárquica que existe entre los diferentes documentos y dentro de ellos. Es decir, debe ser considerada la pirámide legislativa, además de la relación de los artículos. Una vez que se logra la representación normalizada, se puede obtener la temática y términos asociados, apoyándose de algoritmos de clasificación, que hacen uso de técnicas de minería de textos y algoritmos de inteligencia artificial.

El equipo de UPIITA-IPN (UPIITA-IPN, 2019), actualmente evalúa diversas técnicas empleadas para la detección de incoherencias, esto significa que busca la relación entre las definiciones de los términos en el contexto en que son usados. Por ejemplo, si hay dos documentos que regulan la contaminación ambiental, pero uno la define en el ámbito de aguas y otro en el de suelo, entonces las definiciones podrían ser diferentes y no mantener coherencia entre ellas en un contexto general de contaminación. UPIITA e INECOL a través del proyecto integralidad GAMMA llevaron a cabo el Primer Coloquio Interdisciplinario para el Análisis de Legislación Ambiental, con el objetivo de identificar, categorizar y analizar las inconsistencias jurídicas en la legislación ambiental mexicana, así como facilitar el debate sobre la definición de controversias e inconsistencias.

Asimismo, se destaca el proyecto minero La Paila, en el Municipio de Alto Lucero (Aguilar, 2017) mencionado por (Narave & Cházaro, 2017) como un caso real de atropello al medio ambiente debido a actividades mineras. En este proyecto se "plantea la extracción de oro y plata a cielo abierto, por el método de tajo, a través de cortes en el terreno. En nuestro país, la minería es una actividad cuya autorización está reservada a la Federación, de acuerdo con la Constitución Política de los Estados Unidos Mexicanos, a través de la Ley Minera, y en materia ambiental regulada por la Ley General del Equilibrio Ecológico y la Protección al Ambiente (LGEEPA) y sus reglamentos, y de manera particular en el procedimiento de Evaluación del impacto ambiental por la Secretaría de Medio Ambiente y Recursos Naturales (SEMARNAT)".

Otro antecedente fue presentado en el Centro de Investigación en Inteligencia Artificial de la Universidad Veracruzana (Hidalgo, 2018), donde se menciona que la inteligencia artificial "se puede utilizar a favor del medio ambiente, ya que existe una afectación constante hacia la diversidad y a los ecosistemas debido a la intervención humana. Con la finalidad de identificar los impactos cometidos contra el medio ambiente, CONABIO cuenta bases de datos heterogéneas que registran diferentes datos y estos son analizados y preprocesados con el objetivo de obtener conocimiento útil, gracias al uso de técnicas y algoritmos especializados".

Es apropiado preguntar, ¿cómo funciona la inteligencia artificial y si se aplica a los datos ambientales? en respuesta, la primera corresponde a la naturaleza de los datos, es decir, su formato, características y valores; y el segundo corresponde al tipo de modelo requerido por su carácter descriptivo, predictivo o una combinación de ambos. Por ejemplo, una imagen satelital representa una fotografía de alta resolución que se procesa usando una computadora.

Respecto a los documentos de texto, éstos difieren entre sí debido a que los párrafos y los enunciados son de diferente tamaño así como el número de documentos legislativos, México cuenta con leyes generales y federales, reglamentos, normas oficiales mexicanas, constituciones y leyes estatales, entre otras. También cuenta con 70 tratados internacionales centrados en la protección del patrimonio biológico.

Pero ¿por qué prestar atención a dichos documentos? La respuesta parte del hecho de que la legislación ambiental mexicana debe ser revisada y analizada para descubrir sus inconsistencias, sean o no deliberadas, y así proteger a los ecosistemas de riesgos y amenazas. Para lograr este objetivo, el análisis debe aprovechar la minería de texto y el procesamiento del lenguaje natural y técnicas y algoritmos de inteligencia artificial.

“La inteligencia artificial se ha difundido de manera importante hacia todos los sectores desde el ámbito de las finanzas, el transporte, el comercio e incluso como instrumento para el análisis de información que sirve como base para el diseño de las políticas públicas, ya que permite la evaluación precisa de los problemas y necesidades sociales, sirviendo también para la formulación, aplicación y evaluación de políticas” (González et al., 2020).

Con fines de contextualización, la Ley General del Equilibrio Ecológico y la Protección al Ambiente (LGEEPA) contiene normas fundamentales en temas como las áreas naturales protegidas y el orden ecológico, entre otros. Para los expertos, existen deficiencias en los estudios de impacto ambiental de esta ley, y se han evidenciado casos en los que dos leyes, por ejemplo, la LGEEPA y la Ley Minera, entran en conflicto. Sin duda, la tarea es desafiante en varios sentidos: primero, por la gran cantidad de documentos a recopilar y procesar para obtener un corpus ambiental; segundo, realizar la extracción de tópicos permitirá clasificar los documentos; y tercero, y la mayoría importante, desarrollar un análisis semántico no es algo trivial.

Finalmente, los resultados de este trabajo deben traducirse en capacidades que ayuden a los tomadores de decisión a tener acceso a información especializada y oportuna, con la finalidad de incidir en el desarrollo de mejores políticas públicas en materia ambiental.

De acuerdo con lo indagado por el IPN, la legislación mexicana en materia ambiental debe ser revisada y analizada para identificar cualquier inconsistencia, razón por la cual en este trabajo enfocaremos los esfuerzos en el preprocesamiento de dichas leyes.

Cabe destacar que, para dicho preprocesamiento se seleccionaron nueve leyes federales (en formato pdf), cuyo resultado dará origen al corpus que propone este proyecto.

Además, se hará un análisis comparativo de algunas herramientas de software (dedicadas a la minería de textos) con el fin de seleccionar la mejor alternativas para aplicar las tareas de preprocesamiento de textos antes mencionadas. Así mismo, este análisis explora las bibliotecas de *Python* que son apropiadas, las que destacan *NLTK*[3].

[3] *Natural Language Tool Kit, de sus siglas en inglés.*

1.2 IDENTIFICACIÓN DEL PROBLEMA

La problemática por resolver en este proyecto, radica en conocer el comportamiento del tiempo y memoria al momento de ejecutar un proceso de preparación de datos aplicado a textos. Para conseguir esto, este proyecto aprovecha las funcionalidades que proveen tanto herramientas de software como lenguajes de programación *Python*.

Una cifra destacable es que “alrededor de un 80% de la información de las organizaciones está almacenada en forma de texto que no se apega a una estructura determinada” (Solutions, 2021). Una explicación está basada en el aumento de dispositivos y aplicaciones, los cuales generan datos masivos y/o heterogéneos que no se apegan a un estándar de facto (*Big Data*[4]).

En el contexto legislativo, es importante ejecutar dicho proceso de preprocesamiento mediante la identificación de tareas genéricas y específicas basadas en *CRISP-DM* y *TDSP*. Durante su ejecución requerimos medir la memoria y el tiempo transcurridos para las distintas configuraciones de tareas, bajo una infraestructura de hardware determinada (en equipo local y en la nube).

[4] *Se refiere a datos que por su volumen, velocidad y variabilidad son difíciles de procesar y analizar.*

1.3 OBJETIVOS

1.3.1 Objetivo general

Medir los recursos computacionales como tiempo y memoria empleados para realizar tareas de preprocesamiento de documentos de texto, específicamente en la fase de preparación de datos mediante técnicas de minería de textos con la finalidad de caracterizar una dupla (una tarea y un recurso computacional) y de este modo, crear una instancia *EC2* para aprovechar el cómputo en la nube.

1.3.2 Objetivos específicos

- Identificar y ejecutar tareas de pre-procesamiento de textos genéricas y especializadas con el fin de homogeneizar los documentos de leyes ambientales mexicanas (generación de un corpus legislativo) utilizando operadores de minería de textos proporcionados por las herramientas y/o lenguajes de programación específicos.
- Comprender los objetivos del negocio (*business undestanding*) y de los datos (*data understanding*).
- Monitorear los recursos (tiempo y memoria) a partir de diseños experimentales, utilizando *scripts* de *Python* para el preprocesamiento de los datos, es decir, 9 documentos de ley ambiental seleccionados.
- Realizar una o más pruebas escalables con el número de documentos y con las diferentes tareas de preprocesamiento de texto a realizar en la capa gratuita de la máquina virtual de *AWS*.

1.4 JUSTIFICACIÓN

El proyecto que aquí se describe, se llevó a cabo por cada una de las siguientes razones:

1.4.1 Ámbito gubernamental

Los programas nacionales estratégicos (PRONACES) del CONAHCYT son el resultado de una preocupación por conseguir una solución a los distintos problemas que ocurren en México.

El enfoque de esta investigación atiende específicamente al Programa Nacional Estratégicos en Sistemas Socioecológicos y Sustentabilidad (Pronaces-SSyS), el cual tiene como objetivo impulsar la co-producción de conocimiento a nivel técnico-científico, institucional y comunicativo para llevar a cabo acciones de conservación, restauración, uso y aprovechamiento de los ecosistemas, de los recursos naturales y de la biodiversidad desde una perspectiva de sustentabilidad y de justicia social" (Conahcyt, n.d.).

1.4.2 Ámbito académico

Las leyes ambientales mexicanas objeto de estudio son de interés para instituciones como INECOL e IPN, debido principalmente a investigaciones finalizadas sobre big data y procesamiento de texto para la identificación de incongruencias semánticas.

Dentro de los textos del proyecto Integralidad Gamma (Inecol, 2021), se menciona que "las leyes sobre conservación de recursos, se centran generalmente en uno solo de ellos, es decir, sólo en bosques, yacimientos minerales o en animales, o incluso, en algunos otros recursos intangibles tales como zonas circundantes a sitios de alto valor arqueológico. Además, muchas de las leyes que existen para su conservación, no son exclusivamente ambientales, ya que además de contener importantes componentes del medio ambiente integran decisiones de

política ambiental tales como leyes municipales, estatales y nacionales en materia de desarrollo, uso del suelo y de infraestructura".

Con relación al programa de maestría en sistemas computacionales, este trabajo de tesis contribuye a la exploración y aplicación de la mineria de textos en el campo de la sustentabilidad. En particular la LGAC que se atiende es la siguiente:

- Cómputo en la nube: Configuración de la máquina virtual y despliegue de una solución tecnológica en la capa gratuita de *AWS*, utilizando el servicio *EC2*.

Este trabajo contribuye a la solución de problemas en la falta de trabajo en preprocesamiento legislativos, así como la falta de experimentación con bibliotecas de extracción de texto y su respectivo analisis estadístico. Este analisis estadístico arroja resultados relacionados con el consumo de recursos computacionales requeridos para realizar una o más tareas específicas.

Los beneficios que tendrán al finalizar este proyecto son resultados tales como (CNDH, 2018):

- Identificación y análisis de bibliotecas que permitan extraer texto y medir recursos.
- Evaluación de memoria y tiempo en 9 leyes ambientales de carácter federal.
- Comparación de las plataformas de cómputo en la nube como *AWS*, *Microsoft Azure* y *Google Cloud* en su modalidad de capa gratuita.
- Relación de tareas genéricas y específicas de preparación de datos entre *CRISP-DM* y minería de textos.
- Comparación entre operadores de mineria de texto proporcionados por *Python.*
- Presentación de los resultados obtenidos en cada uno de los casos de pruebas desde una perspectiva estadística.

1.5 ALCANCES Y LIMITACIONES

El alcance y las limitaciones de este proyecto forman parte de su justificación, es decir, de la explicación contextual de su importancia, con base en las cuales se definen las expectativas del mismo.

1.5.1 Alcances

- Se abordarán las tres primeras fases del modelo de proceso *CRISP-DM*, es decir, las fases de entendimiento del negocio, entendimiento de los datos y preparación de los datos.
- Utilizar *Python* con sus bibliotecas, como herramienta de minería de textos y/o preprocesamiento.
- Analizar 9 leyes ambientales mexicanas de ámbito federal, descritas en el capítulo de marco teórico utilizadas en el diseño experimental.
- Los recursos computacionales de interés son el tiempo y memoria *RAM*.
- Las características técnicas de la computadora donde se monitorearán los recursos (tiempo y memoria) Memoria *RAM* de 3 GB. Disco duro de 500 GB. Procesador AMD 1.60 Ghz x64. Windows 8.1 Pro.

1.5.2 Limitaciones

- Utilización de una instancia *EC2* de *AWS* en modalidad de capa gratuita.
- Incompatibilidad en bibliotecas para su ejecución en la nube.
- Tiempo en la utilización de la instancia, la cual puede causar costos no deseados.

1.6 PREGUNTAS DE INVESTIGACIÓN

Derivado de la contextualización anterior, surgen las siguientes preguntas de investigación:

- ¿En qué consiste el proceso de transformación del formato original en pdf a texto plano realizado por cada una de las bibliotecas de *Python*?
- ¿Cuál es el tiempo que toma la extracción de texto de un documento en formato original en pdf y su copiado/escritura en un archivo en formato texto plano?
- ¿Qué elementos dificultan la transformación de la tarea de formato original en pdf a texto plano?
- ¿Qué tipo de correlación existe entre el número de leyes y el número de tareas de preprocesamiento y cómo es el comportamiento de tiempo y la memoria en cada caso?
- ¿Qué resultados se obtienen al ejecutar las tareas de preprocesamiento en *Python*?
- ¿Qué diferencia se existe entre el consumo de tiempo y memoria respecto al número de páginas, número de palabras, número de encabezados, entre otras, en cada documento de ley?

CAPÍTULO II. MARCO TEÓRICO

En el presente capítulo se hará una recopilación de las consideraciones teóricas que sustentan este proyecto de investigación, con el fin de sentar las bases que guíen el planteamiento y propuesta de solución del mismo.

2.1 MINERÍA DE TEXTOS

“La minería de textos es un proceso que permite descubrir conceptos clave y relaciones entre dichos textos, la cual tiene aplicaciones en muchos campos, como la economía nacional, la gestión social, los servicios de información y la seguridad, entre otros” (Feldman & Sanger, 2007). Dicho de otra manera, “la minería de textos puede ser de utilidad para predecir la situación económica y las tendencias del mercado, para descubrir oportunamente factores de inestabilidad social, para identificar fenómenos e irregularidades en materia de salud” (Qingkai et al., 2020), e incluso, para analizar los fundamentos jurídicos que rigen la interpretación y aplicación de las leyes, tema que compete a este estudio.

Con el propósito de comprender el proceso que lleva a cabo la minería de textos, se definirá este término, su evolución, importancia, etapas y clasificación.

2.1.1 Definiciones de minería de textos

“La minería de textos se entiende como el proceso de extraer el conocimiento implícito de datos textuales. Dicho conocimiento, debe distinguirse de la información almacenada previamente mediante tareas tales como: clasificar, agrupar y asociar” (Feldman & Sanger, 2007). El texto se refiere a un conjunto ordenado de párrafos, lo cual no incluye palabras escritas en un lenguaje artificial, como el código fuente o las ecuaciones matemáticas.

Cabe mencionar, que la minería de textos no es un concepto nuevo, es sólo una clasificación de la minería de datos y, por lo tanto, todos los algoritmos de esta área pueden aplicarse sobre los datos textuales. La diferencia entre los dos consiste en que la minería de datos se aplica a datos estructurados y relacionales, mientras que la minería de textos se ocupa de todos los elementos no estructurados y semiestructurados.

Dado lo anterior, a continuación se muestran los elementos que la caracterizan.

2.1.2 Evolución de la minería de textos

La minería de textos, como campo de investigación que cruza múltiples tecnologías, se originó en técnicas únicas de clasificación, agrupación de texto y resumen automático de texto. En la década de los años 50, la clasificación y agrupación de textos surge como una aplicación de reconocimiento de patrones. En ese momento, la investigación se centró en la clasificación y agrupamiento de la información contenida en libros. (Luhn, 1958) propuso el concepto de resumen automático, refiriéndose a la condensación de información automática, lo cual agregó valor al campo de la minería de textos.

En los últimos años, la capacidad de generar y recolectar datos ha aumentado significativamente, debido al gran poder de procesamiento de las computadoras, así como al bajo costo de almacenamiento.

A continuación, en la Tabla 2, se muestran las técnicas de minería de textos surgidas a través de su evolución.

Tabla 2. Técnicas de minería de textos (Tandel et al., 2019).

Extracción de información	Etapa principal que consiste en reconocer las frases importantes dentro del texto, extrayendo información significativa de grandes trozos de texto en forma de registros para su posterior acceso o recuperación.
Recuperación de información	Su objetivo es obtener el documento con información precisa recuperada por el usuario tras la solicitud enviada por el usuario.
Resumen	Se toma el texto en bruto y se realizan en él operaciones de preprocesamiento, aplicando tres métodos: la *tokenización,* el *stemming* y la eliminación de palabras.
Agrupación *(Clustering)*	Es un proceso de seccionar un grupo de objetos o datos en una colección de subclases relevantes y comprensibles. Éste se utiliza principalmente para hacer un conjunto de documentos y archivos similares.
Categorización	Se reconocen los temas importantes de un documento. Esto se hace asignando los documentos a un conjunto de temas predefinidos. El documento categorizado puede tratarse como una "bolsa de palabras".

Si bien estas técnicas se aplican en la fase de minería de textos, existen otras dos fases, una previa y otra posterior, las cuales se explican a continuación (Taeho, 2019):

- Preprocesamiento: Se refiere a la realización de operaciones o transformaciones sobre el texto en algún tipo de representación estructurada o semiestructurada que facilite su posterior análisis. Algunas de las técnicas utilizadas para la transformación de documentos en una forma intermedia pueden ser: análisis de texto, categorización, técnicas de procesamiento de lenguaje natural (etiquetado de parte del discurso, tokenización, lematización), técnicas de extracción de información (categorización, adquisición de patrones léxico sintáctico, extracción automática de términos, localización de trozos específicos de texto), así como técnicas de recuperación de información (indexación).

- Visualización de los resultados: Exploración guiada de los datos para que sea lo más amigable posible para el usuario. Las últimas tendencias presentan los resultados mediante el uso de páginas web, gráficos y *dashboards*.

El interés en descubrir información desconocida radica en que la minería de textos depende no sólo del contenido de éste, sino también de la búsqueda de su contexto. En este sentido, se requiere proporcionar el entorno para la recopilación de patrones similares de manera conjunta, así como realizar tareas de agrupación, visualización y navegación, para conocer la relación entre los patrones, descubrir nuevas relaciones y resumir el texto, lo que conlleva a analizar el contenido mediante técnicas de clasificación.

Los datos deben convertirse a un formato semiestructurado o formato estructurado para poder aplicar los algoritmos de aprendizaje automático de minería de datos fácilmente. Esta conversión de datos se realiza mediante el preprocesamiento de los mismos, cuyas técnicas se explican en el siguiente apartado.

2.2 TÉCNICAS DE PRE-PROCESAMIENTO DE TEXTOS

El preprocesamiento pretende obtener un conjunto de datos que sea útil para la fase de extracción de conocimiento. Las técnicas específicas utilizadas en este proyecto son:

a) Transformación del formato original en pdf a formato de texto plano.

b) Conversión del formato de texto plano (txt) a formato json[5].
c) Identificación de entidades.
d) Etiquetado de textos.

Las tareas a y b pertenecen a la categoría de limpieza, transformación y formateo según *CRISP-DM*, así como c y d pertenecen a la categoría de construcción e integración según *CRISP-DM*.

A continuación, se describen cada una de estas técnicas:

2.2.1 Transformación del formato original en pdf a formato de texto plano

"Los textos se transforman en una representación estructurada o semiestructurada que facilite su posterior análisis. Se define el conjunto corpus de documentos, los cuales deben ser representativos y seleccionarse aleatoriamente o mediante algún método de muestreo probabilístico. Asimismo, debe evitarse en esta etapa, la duplicación de documentos dentro del corpus" (Cortez, 2018).

"La *tokenización* se define como el proceso de segmentar un texto o textos en *tokens* por el espacio en blanco o los signos de puntuación. Ésta se puede aplicar tanto a códigos fuente en cualquier lenguaje de programación, así como a textos escritos en lenguaje natural" (Aho et al., 2007).

El proceso de *tokenizar* un texto representado en la Figura 2, indica que un texto dado está dividido en *tokens* por un espacio en blanco, signos de puntuación y caracteres especiales. Las palabras que incluyen uno o algunos de los caracteres especiales, se eliminan. El primer caracter de cada oración se da como el caracter en mayúsculas, por lo que debe ser cambiado a minúsculas. Las palabras redundantes deben eliminarse después de los pasos de indexación de texto.

[5] *JavaScript Object Notation.*

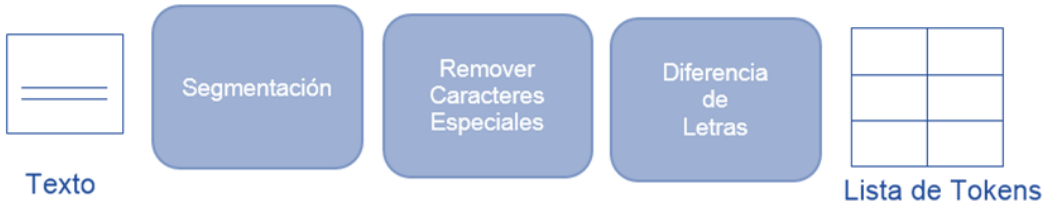


Figura 2. Proceso de tokenizar texto (Taeho, 2019).

En el ejemplo consistente en *tokenizar* el texto (Figura 3), éste consta de dos oraciones, las cuales constituyen el texto que se segmenta en *tokens* por el espacio en blanco. Se muestra contenido en el Anexo 5.

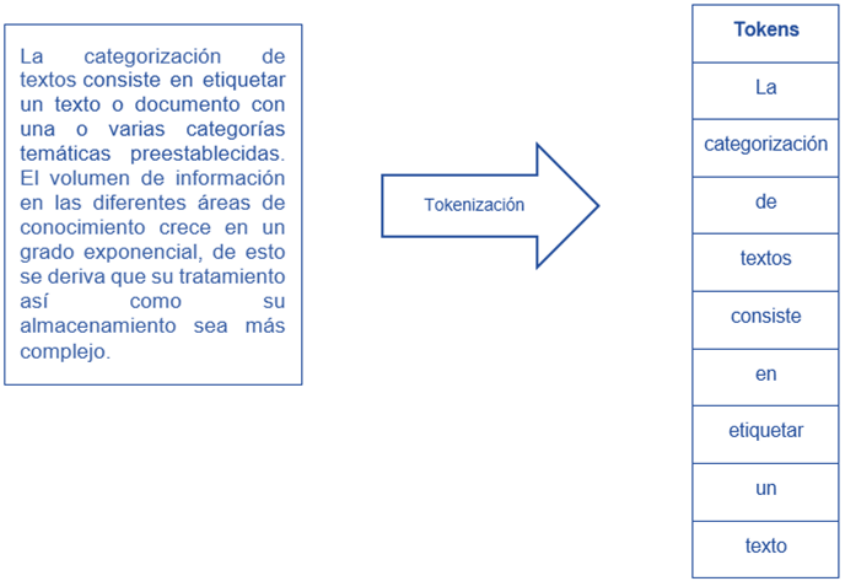


Figura 3. Ejemplo de tokenizar texto.

2.2.2 Conversión del formato de texto plano (txt) a formato json

Se convierte el texto resultante de la tarea anterior a formato json. Para la evaluación manual, se consideran: caracteres especiales, tamaño del documento, estilo de las páginas, texto en imágenes, texto en tablas y tiempo de ejecución.

Se utiliza *Apache Tika* para transformar documentos en texto plano, se trata de un marco de análisis y detección de contenido escrito en *Java* que contiene un conjunto de herramientas que detectan y extraen metadatos y contenido de texto en formatos como: word, pdf, entre otros, se obtiene el texto del documento, incluido el texto en tablas e imágenes; los encabezados y pies de página repetidos tantas veces como páginas. Se implementaron *scripts* en *Python* para identificar y tratar: encabezados y pies de página, tablas, imágenes, números de página y notas de modificación, manteniendo sólo los textos. Fue el paso más difícil debido a la diversidad de documentos y a los caracteres.

Se realiza un estudio de la estructura jerárquica interna de las leyes con el fin de obtener una representación del documento que redujera al máximo la pérdida de información. El análisis demostró que las leyes no mantienen una estructura uniforme, sin embargo, al analizarlos manualmente, se dedujo una estructura generalizada de las leyes en formato json, el cual puede apreciarse en el Anexo 6.

En dicha estructura se integraron y unificaron en archivos tipo json todos los posibles campos que el documento pudiera contener o no. Lo anterior facilita la representación de los datos por niveles, de tal forma que se puede interpretar que los títulos contienen capítulos, que a su vez contienen artículos.

En cuanto a la estructura cada ley tiene un nombre y una descripción general. Las leyes tienen divisiones donde el título y los artículos transitorios fueron considerados como el segundo nivel. Cada título tiene un número y un nombre. Los capítulos suelen estar dentro del título, pero a veces no. Dentro del capítulo, podemos encontrar artículos que son la división fundamental de las leyes. A

continuación se describe con más detalle cada uno de los elementos de esta deconstrucción jerárquica de las mismas.

2.2.3 Identificación de entidades

“El reconocimiento de entidades con nombre (*NER*), también conocido como extracción de entidades, consiste en localizar y clasificar partes del texto estudiado en categorías preestablecidas como lugares, personas, organizaciones, expresiones de tiempo y cantidades” (Martinez-Seis et al., 2022).

Para realizar el proceso de reconocimiento de entidades se utiliza *spaCy*, el cual proporciona una serie de anotaciones lingüísticas sobre la estructura gramatical de un texto, como los tipos de palabras, las partes de la oración, el análisis morfológico, la lematización, el reconocimiento de entidades con nombre y el análisis sintáctico de dependencias.

Esta biblioteca funciona con redes neuronales convolucionales y proporciona modelos pre entrenados de distintos idiomas; además, permite crear nuevos modelos o reentrenar los modelos proporcionados con datos propios para crear modelos en campos específicos. Puede reconocer varios tipos de entidades con nombre en un documento, pidiendo al modelo una predicción. Los modelos son estadísticos y dependen en gran medida de los ejemplos con los que se han entrenado. Tiene una base de entidades con nombre en diferentes idiomas, incluido el español. Se tienen entidades en el texto y se sustituyeron como un solo término. Por ejemplo, Poder Judicial de la Federación es una entidad porque representa una instancia: tiene cinco palabras en español y su entidad podría ser poder_judicial_de_la_federación.

2.2.4 Etiquetado de textos

“El proceso de lematización permite obtener la forma canónica de un conjunto de variantes morfológicas. En esta tarea se utiliza *FreeLing* para el proceso de lematización, herramienta que incluye diccionarios lingüísticos para el español y otras lenguas. Estos diccionarios se obtienen de diferentes proyectos externos de código abierto. El proceso de etiquetado incluye el preprocesamiento de cada elemento de la estructura mediante algoritmos para obtener la lematización, el etiquetado de parte del habla (*POS*) y la frecuencia de términos (*TF*)” (Martinez-Seis et al., 2022).

El etiquetado *POS* (*Part-of-Speech*) es un proceso de procesamiento del lenguaje natural que consiste en clasificar las palabras de un texto (*corpus*) en correspondencia con una parte de la oración determinada, en función de la definición de la palabra y de su contexto. El proceso de etiquetado puede llevarse a cabo utilizando la biblioteca *spaCy* o *Freeling* la cual es una biblioteca *C++* que proporciona funcionalidades de análisis lingüístico, como análisis morfológico, detección de entidades con nombre, etiquetado *POS*, análisis sintáctico, desambiguación del sentido de las palabras, etiquetado de roles semánticos, etc., para diferentes idiomas (inglés, español, portugués, italiano, francés, alemán, ruso, catalán, gallego, croata, esloveno, etc.).

La frecuencia de términos es la medida de la frecuencia con que una palabra o entidad aparece en un documento. Se calcula la frecuencia de términos (*TF*) de cada documento antes y después de la detección de entidades. A continuación, para cada elemento de la estructura (título, capítulo o artículo) se ofrece una lista de palabras/entidades con su respectivo *TF*.

2.3 USOS Y APLICACIONES DE LA MINERÍA DE TEXTOS

La minería de textos tiene aplicación en diferentes campos, como la medicina, la biología, el análisis de opiniones y la gestión documental, entre otras, así como el caso de este proyecto, en el que se analizan las leyes ambientales.

De manera general, en las disciplinas antes mencionadas, se aplica para diversas funciones, tales como:

a) Extracción de información.
b) Análisis de sentimientos o minería de opiniones.
c) Clasificación documental.
d) Elaboración de resúmenes.
e) Extracción de conocimiento.
f) Procesamiento del lenguaje natural.

“La minería de textos está relacionada con el Procesamiento del Lenguaje Natural (*PNL*), que incluye técnicas de inspiración lingüística, es decir, un texto se analiza típicamente desde un punto de vista léxico y sintáctico utilizando una gramática formal, la información resultante se interpreta semánticamente y se utiliza para extraer información sobre lo dicho” (Kao & Poteet, 2007).

2.3.1 Procesamiento del lenguaje natural

Las tecnologías basadas en el *PNL* se están generalizando cada vez más. Por ejemplo, los teléfonos y las computadoras de mano admiten el reconocimiento predictivo de texto y escritura a mano; los motores de búsqueda web dan acceso a información contenida en texto no estructurado. Proporciona interfaces hombre-máquina más natural y un acceso más sofisticado a información almacenada, el procesamiento del lenguaje ha llegado a desempeñar un papel central en la sociedad de información.

"*NLTK* define una infraestructura que se puede utilizar para crear programas de *PNL* en *Python*. Eso proporciona clases básicas para representar datos relevantes para el procesamiento del lenguaje natural; interfaces estándar para realizar tareas como etiquetado de parte de la voz, análisis sintáctico, y clasificación de textos; e implementaciones estándar para cada tarea que se puede combinados para resolver problemas complejos. Se centran en las palabras: cómo identificarlas, analizar su estructura, asignar ellos a las categorías léxicas, y acceden a sus significados" (Bird et al., 2009).

2.3.2 Clasificación de textos en redes sociales (social media)

"En los últimos años las redes sociales han jugado un papel muy importante en la comunicación. *Twitter* ha sido una compañía sobresaliente en este medio, gracias a la implementación del concepto de microblogging, que consiste en publicaciones de textos cortos con una longitud máxima de 140 caracteres. Existe una extensa cantidad de tweets públicos que circulan en la red social que no están categorizados en un tópico específico, limitando la explotación de dicha información, por ejemplo, una clasificación orientada con un objetivo específico nos podría ayudar a medir el grado de aceptación de un producto, servicio, líder político, etc" (Fidencio et al., 2017).

La clasificación automática de textos se refiere a la actividad de etiquetar textos de lenguaje natural en categorías específicas, mediante el uso de sistemas computacionales. Este proceso de clasificación consta de tres etapas: pre-procesamiento de los documentos, la construcción del clasificador y categorización de nuevos documentos. Se enfoca en el estudio, descripción e implementación de las técnicas utilizadas en la etapa de pre procesamiento de datos aplicables a *tweets* en español.

2.3.3 Categorización de documentos

La extracción de información fue implementada utilizando selectores de *jQuery*, para la obtención de los elementos que contienen la información. En el caso de realizar la extracción de texto en otras páginas web, deberá modificarse el programa para adaptarse a las nuevas características y disposición que tomen los elementos que contengan la información, siendo esto incómodo y poco óptimo.

Sin embargo, es la única forma de extraer información de una página, servicio, o aplicación web que no disponga de una *API* (*Application Programming Interface*) que presente la información de manera independiente a la vista o interfaz de usuario. Uno de los problemas más importantes fue el exceso de clases en comparación con el número de instancias. Muchas de estas clases contenían a un único elemento, lo que dificultaba el proceso de clasificación al no obtener suficiente información determinante para caracterizar cada tipo de clase“ (Hernández, 2016).

Se considera la extracción de texto con el fin de obtener una cantidad suficiente de palabras para caracterizar cada una de las instancias a clasificar. Por otra parte, se tiene en cuenta la posibilidad de descargar datos de diferentes disciplinas a la Informática, para aumentar la diferenciación entre instancias y esperando que la capacidad de clasificación incremente de manera considerable.

2.3.4 Búsqueda de fundamentos jurídicos

La forma en que los juristas han sobrellevado estas dificultades ha sido la especialización en el estudio de las distintas ramas jurídicas; sin embargo, esto ha limitado el análisis integral del sistema jurídico y genera dificultades al abordar problemas que involucran diversas áreas legales.

Por otra parte, un elemento que caracteriza al ámbito legal es que la mayor parte de la información que en él se genera se encuentra en forma de texto, constituyendo las sentencias judiciales un componente sustancial de esta

información. El cual consiste en detectar los roles semánticos en una norma. Se toma como punto de partida el texto de una disposición jurídica, así como la clase que corresponde a dicha disposición. El sistema se encarga de asignar las etiquetas semánticas que le corresponden a los elementos de la disposición como son: destinatario de la norma, verbo rector de la norma y disposición jurídica en concreto.

"En la etapa de pre-procesamiento fueron etiquetadas las categorías gramaticales de las palabras (*POS tagging*) y posteriormente se hizo un análisis sintáctico superficial (*chunks*). A partir de este análisis sintáctico se localizan elementos relevantes de las normas tales como verbos, sujetos, oraciones y frases. Una vez localizados estos elementos se asignó una plantilla a la norma, dependiendo de la clase de norma jurídica que le correspondía. Finalmente los espacios en la plantilla se cubrieron con los elementos identificados en el análisis sintáctico" (Sandoval, 2016).

Sin embargo, la aplicación que compete a este estudio se detalla en el siguiente apartado.

2.4 MINERÍA DE TEXTOS EN LEGISLACIÓN AMBIENTAL

La aplicación de la minería de textos para este proyecto de investigación hace referencia al área de legislación ambiental, es decir, al conjunto de documentos que pretenden regular la interacción de la humanidad y el medio ambiente natural. Para ello se empezará por definir cómo está constituida la jerarquía normativa mexicana.

2.4.1 Normativa mexicana

Según el artículo 133 de la Constitución Política De Los Estados Unidos Mexicanos, (Galvan, 2005) define que: "las Leyes del Congreso de la Unión que emanen de ella, y todos los Tratados hechos y que se hicieren por el Presidente de la República, con aprobación del Congreso, serán la Ley Suprema de toda la Unión. Los jueces de cada Estado se ajustarán a dicha Constitución, Leyes y Tratados, a pesar de las

disposiciones en contra que pueda haber en las Constituciones o Leyes de los Estados."

Dado lo anterior, la jerarquía del orden jurídico en el Derecho mexicano es la siguiente (Guzman Monter, 2017):

- La Constitución: Cada estado es autónomo y tiene su propia constitución.
- Los Tratados Internacionales: Acuerdos entre dos o más estados soberanos, firmados por el presidente y ratificados por el senado.
- Leyes Federales: Son creadas por el congreso de la unión y determinadas por la Constitución, la propia ley nacional y otras leyes que incidan en la materia.
- Las Leyes Ordinarias: Son las leyes que crean los congresos locales de cada entidad federativa. También se les conoce como leyes estatales o del fuero común.
- Los Decretos: Acto administrativo emanado del poder ejecutivo y que tienen un carácter normativo reglamentario aplicable a todas las personas cuya situación recae bajo el campo de acción.
- Los Reglamentos: Norma jurídica de carácter general dictada por la Administración pública y con valor subordinado a la Ley.
- Las Normas Jurídicas Individualizadas: Regulan situaciones específicas sobre uno o más sujetos en particular. Contratos, convenios, testamentos, resoluciones.

Aunado a la clasificación anterior, se va a acotar al ámbito de leyes ambientales, las cuales, “establecen las normas para la conservación, protección, mejoramiento y restauración del medio ambiente y los recursos naturales que lo integran” (Mejía, 2000).

2.4.2 Legislación ambiental

Las leyes ambientales seleccionadas para el presente proyecto, todas del ámbito federal y en formato pdf, se muestran en la Tabla 3, identificando su propósito.

Tabla 3. Leyes ambientales con su función.

Ley Ambiental	Función
LGEEPA: Ley General del Equilibrio Ecológico y la Protección al Ambiente	"Preservar y restaurar del equilibrio ecológico, así como a la protección al ambiente, en el territorio nacional y las zonas sobre las que la nación ejerce su soberanía y jurisdicción" (Congreso general de los estados Unidos Mexicanos, 2015).
LAN: Ley de Aguas Nacionales	"Regular la explotación, uso o aprovechamiento de dichas aguas, su distribución y control, así como la preservación de su cantidad y calidad para lograr su desarrollo integral sustentable en materia de aguas nacionales; es de observancia general en todo el territorio nacional, sus disposiciones son de orden público e interés social" (Congreso general de los estados Unidos Mexicanos, 2020a).
LDRS: Ley de Desarrollo Rural Sustentable	"Integrar una política de Estado para el desarrollo rural, por encima de las naturales diferencias entre las fuerzas políticas, capaz de construir acuerdos en puntos básicos que garanticen metas y programas en el largo plazo, creadora de seguridad, confianza y certidumbre; como una de las principales aspiraciones de los productores y sus organizaciones" (Congreso general de los estados Unidos Mexicanos, 2007).
LGPAS: Ley General de Pesca y Acuacultura Sustentable	"Ordenar, fomentar y regular el manejo integral y el aprovechamiento sustentable de la pesca y la acuacultura, considerando los aspectos sociales, tecnológicos, productivos, biológicos y ambientales" (Congreso general de los estados Unidos Mexicanos, 2018a).
LGVS: Ley General de Vida Silvestre	"Regular el aprovechamiento sustentable de los recursos forestales maderables y no maderables y de las especies cuyo medio de vida total sea el agua, será regulado por las leyes forestales y de pesca, respectivamente, salvo que se trate de especies o poblaciones en riesgo" (Congreso general de los estados Unidos Mexicanos, 2018b).
LGDFS: Ley General de Desarrollo Forestal Sustentable	"Promover la legalidad en las actividades productivas, mejorar la capacidad de transformación e integración industrial, impulsar la comercialización y fortalecer la organización de redes locales de valor y cadenas productivas del sector forestal" (Mexicanos, 2018).

LGPGIR: Ley General Para la Prevención y Gestión Integral de Residuos	“Garantizar el derecho de toda persona al medio ambiente sano y propiciar el desarrollo sustentable a través de la prevención de la generación, la valorización y la gestión integral de los residuos peligrosos, de los residuos sólidos urbanos y de manejo especial; prevenir la contaminación de sitios con estos residuos” (Mexicanos, 2021).
LFBOGM: Ley Federal de Bioseguridad de Organismos Genéticamente Modificados	“Regular las actividades de utilización confinada, liberación experimental, liberación en programa piloto, liberación comercial, comercialización, importación y exportación de organismos genéticamente modificados, con el fin de prevenir, evitar o reducir los posibles riesgos que estas actividades pudieran ocasionar a la salud humana o al medio ambiente y a la diversidad biológica o a la sanidad animal, vegetal y acuícola” (Mexicanos, 2020).
LGCC: Ley General De Cambio Climático	“Garantizar el derecho a un medio ambiente sano y establecer la concurrencia de facultades de la federación, las entidades federativas y los municipios en la elaboración y aplicación de políticas públicas para la adaptación al cambio climático y la mitigación de emisiones de gases y compuestos de efecto invernadero” (Congreso general de los estados Unidos Mexicanos, 2020b).

Una vez comprendidos los temas referentes a la legislación mexicana, en específico la ambiental, así como su relación con la minería de textos, se describen los tópicos relacionados con las tecnologías a utilizar para dicho fin.

2.5 HERRAMIENTAS TECNOLÓGICAS Y ESTADÍSTICAS

Las herramientas tecnológicas se refieren a todo aquel *software* o *hardware* cuyo objetivo es contribuir al desarrollo del proyecto.

2.5.1 Python

“Python es un lenguaje de alto nivel de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código, se utiliza para desarrollar aplicaciones de todo tipo, ejemplos: Instagram, Netflix, Spotify, Panda3D, entre otros. Está desarrollado bajo una licencia de código abierto aprobada por OSI (modelo de interconexión de sistemas abiertos) lo que lo hace de libre uso y distribución, incluso para uso comercial. La licencia es administrada por *Python Software Foundation”* (Python, 2021).

Las bibliotecas específicas utilizadas son las siguientes:

a) PyPDF2

"PyPDF2 es una biblioteca de *Python* de código abierto y gratuita capaz de dividir, fusionar, recortar y transformar las páginas de los archivos pdf. Ésta puede agregar datos personalizados, opciones de visualización y contraseñas a archivos pdf, así como recuperar texto y metadatos" (PyPDF2, 2022).

b) Pdfplumber

"*Pdfplumber* es una biblioteca gratuita de *Python* de código abierto capaz de extraer texto de cualquier página dada. También puede intentar preservar el diseño de un texto, así como identificar las coordenadas de las palabras" (Github, 2022).

c) Json

"Python tiene un paquete integrado llamado *JavaScript Object Notation (JSON)*, el cual es un formato estandarizado que se utiliza habitualmente para transferir datos en forma de texto que pueden enviarse a través de una red" (W3schools, 2022).

d) Apache Tika

"Es una biblioteca que se utiliza para la detección del tipo de documento y la extracción de contenido de varios formatos de archivo, así como marco de análisis y detección de contenido. Detecta y extrae metadatos y texto de más de mil tipos de archivos diferentes" (Pypi, 2022).

e) Pandas

Es una biblioteca de *Python* especializada en el manejo y análisis de estructuras de datos (Sanchez, 2021b):

- Define nuevas estructuras de datos basadas en los arrays de la biblioteca *NumPy* pero con nuevas funcionalidades.
- Permite leer y escribir fácilmente ficheros en formato CSV, Excel y bases de datos *SQL*.
- Permite acceder a los datos mediante índices o nombres para filas y columnas.
- Ofrece métodos para reordenar, dividir y combinar conjuntos de datos.
- Permite trabajar con series temporales.
- Realiza todas estas operaciones de manera muy eficiente.

f) Numpy

"Es una biblioteca de *Python* especializada en el cálculo numérico y el análisis de datos, especialmente para un gran volumen de datos que incorpora una nueva clase de objetos llamados arrays que permite representar colecciones de datos de un mismo tipo en varias dimensiones, y funciones muy eficientes para su manipulación, de una dimensión (vector), de dos dimensiones (matriz), de tres dimensiones (cubo), y así sucesivamente, no habiendo límite en el número de dimensiones del array más allá de la memoria disponible en el sistema" (Sanchez, 2021a).

g) NLTK

"NLTK (*Natural Language Toolkit, de sus siglas en inglés*) es una plataforma líder para crear programas de *Python* para trabajar con datos de lenguaje humano. Proporciona interfaces fáciles de usar para más de 50 corpus y recursos léxicos como WordNet, junto con un conjunto de bibliotecas de procesamiento de texto para clasificación, tokenización, lematización, etiquetado, análisis y razonamiento semántico, contenedores para bibliotecas de procesamiento del Lenguaje Natural (*NLP)* y un foro de discusión activo" (NLTK, 2021).

NLTK además, define una infraestructura que se puede utilizar para crear programas de *PNL* en *Python*. Eso proporciona clases básicas para representar datos relevantes para el procesamiento del lenguaje natural; interfaces estándar para realizar tareas como etiquetado de parte de la voz, análisis sintáctico, y clasificación de textos; e implementaciones estándar para cada tarea que se puede combinados para resolver problemas complejos. Se centran en las palabras: cómo identificarlas, analizar su estructura, asignar ellos a las categorías léxicas, y acceder a sus significados.

Por otra parte, se hace uso de los paquetes informáticos que se exponen a continuación:

2.5.2 R Project

R Project es un entorno de software libre para computación estadística y gráficos. Compila y se ejecuta en una amplia variedad de plataformas UNIX, Windows y MacOS.

El análisis estadístico se llevó a cabo utilizando el software *R Project* versión 4.2.2, del cual se utilizaron las siguientes herramientas:

a) Diagrama de cajas y alambres

“Representa gráficamente una serie de datos numéricos a través de sus cuartiles. De esta manera, se muestran a simple vista la mediana y los cuartiles de los datos, y también pueden representarse sus valores atípicos” (Jmp, 2022).

b) Diagrama de dispersión

“Usa una colección de puntos colocados usando coordenadas cartesianas para mostrar valores de dos variables. Al mostrar una variable en cada eje, se puede detectar si existe una relación o correlación entre las dos variables. Se pueden interpretar varios tipos de correlación a través de los patrones mostrados en los

diagramas de dispersión. Estos son: positivo (los valores aumentan juntos), negativo (un valor disminuye a medida que el otro aumenta), nulo (sin correlación), lineal, exponencial y en forma de U. La fuerza de la correlación puede determinarse por la proximidad de los puntos entre sí en el gráfico. Los puntos que terminan muy lejos del conjunto general de puntos se conocen como valores atípicos" (Catalogue, 2022).

c) Método Wilcoxon

"El test no paramétrico prueba de los rangos con signo de Wilcoxon, permite comparar poblaciones cuando sus distribuciones (normalmente interpretadas a partir de las muestras) no satisfacen las condiciones necesarias para otros test paramétricos. Es una alternativa al t-test de muestras dependientes cuando las muestras no siguen una distribución normal (muestran asimetría o colas) o cuando tienen un tamaño demasiado reducido para poder determinar si realmente proceden de poblaciones normales" (Datos, 2016).

Este método se utiliza si el tamaño de las muestras es suficientemente grande para determinar (por métodos gráficos o contrastes de hipótesis) que la distribución de las poblaciones a comparar no es de tipo normal, en tal caso, los t-test no son adecuados, por lo que mejor emplear Wilcoxon.

Finalmente, dado que el estudio pertenece a la línea de investigación de cómputo en la nube, en el siguiente apartado, se dan a conocer los temas relacionados con este rubro.

2.6 CÓMPUTO EN LA NUBE

Otro de los temas que compete a esta investigación, se refiere al análisis de los recursos computacionales tiempo y memoria, para lo cual se utiliza el cómputo en la nube.

"La computación en nube ofrece a las empresas modelos prácticos para acceder a las ofertas de infraestructura, plataforma y software de pago por uso. Con la computación en nube, las empresas pueden liberar capital, optimizar el mantenimiento de *TI*, modernizar y escalar los enfoques empresariales, convertir la seguridad y la flexibilidad en servicios y soluciones, ayudar a los clientes de nuevas maneras, y hacer crecer su empresa en las condiciones de mercado siempre cambiantes" (Cisco, 2021a).

2.6.1 Tipos de cómputo en la nube

Los tipos de cómputo en la nube son (Cisco, 2021c):

- Nube privada: Se refiere a los recursos de computación en nube que se usan exclusivamente en una misma empresa u organización. La nube privada puede almacenarse en el centro de datos interno o por medio de un proveedor de servicios.
- Nube pública: Toda la infraestructura de hardware, software y soporte es propiedad del proveedor de servicios que la administra y la proporciona exclusivamente por Internet. Puede acceder a estos servicios y administrar la cuenta mediante un navegador web.
- Nube híbrida: Combina las nubes públicas y privadas para compartir datos y aplicaciones. Conectan la infraestructura y las aplicaciones entre recursos en nube con los recursos existentes que no se encuentran en la nube.

2.6.2 Servicios

"Los servicios de computación en nube ofrecen modelos convenientes de pago por uso que eliminan los gastos y el mantenimiento costosos. Los proveedores de nube alojan una variedad de ofertas de infraestructura, plataforma y software en las instalaciones que ellos "alquilan", lo que le aporta a la organización la flexibilidad de activar o desactivar los servicios de computación en nube en función de los requisitos cambiantes" (Cisco, 2021b):

- Infraestructura como servicio (*IaaS*): En este modelo, un proveedor de nube aloja los componentes de infraestructura que tradicionalmente se almacenan en centros de datos internos. Su organización permite elegir cuándo y cómo desean administrar las cargas de trabajo sin necesidad de comprar, administrar y respaldar la infraestructura subyacente. IaaS permite que la infraestructura esté en funcionamiento rápidamente, con un modelo de pago por uso.
- Plataforma como servicio (*PaaS*): Proporciona componentes de infraestructura para alojar y administrar sistemas operativos y middleware que los desarrolladores necesitan para crear y ejecutar aplicaciones. PaaS ofrece un modelo a petición de pago por uso.
- Software como servicio (*SaaS*): Aloja y administra toda la infraestructura, además de las aplicaciones para usuarios finales. SaaS está disponible a petición o por suscripción.

2.6.3 Proveedores de servicios

a) AWS

"Amazon Web Services (*AWS*) es la plataforma en la nube más adoptada y completa en el mundo, que ofrece más de 200 servicios integrales de centros de datos a nivel global. Millones de clientes, incluso las empresas emergentes que

crecen más rápido, las compañías más grandes y los organismos gubernamentales líderes, están usando *AWS* para reducir los costos, aumentar su agilidad e innovar de forma más rápida" (AWS, 2021).

AWS tiene la infraestructura más extensa del mundo, es proveedor de servicios en la nube ofrece tantas regiones con diferentes zonas de disponibilidad unidas por redes de baja latencia, alto rendimiento y alta redundancia. Cuenta con 84 zonas de disponibilidad repartidas en 26 regiones geográficas de todo el mundo, agregando 24 zonas de disponibilidad adicionales y 8 regiones adicionales en Australia, Canadá, India, Israel, Nueva Zelanda, España, Suiza y los Emiratos Árabes Unidos (EAU). Gartner ha respaldado los modelos de región y zona de disponibilidad de *AWS* como la forma recomendada de ejecutar aplicaciones para empresas que requieren alta disponibilidad.

b) Microsoft Azure

"La plataforma *Azure* está compuesta por más de 200 productos y servicios en la nube diseñados para ayudarle a dar vida a nuevas soluciones que permitan resolver las dificultades actuales y crear el futuro. Cree, ejecute y administre aplicaciones en varias nubes, en el entorno local y en el perímetro, con las herramientas y los marcos que prefiera" (Azure, 2021).

Azure es la única nube híbrida proporcionando una mayor productividad de desarrollo, seguridad completa en varias capas y un mayor nivel de cumplimiento normativo que cualquier otro proveedor de la nube. Además, Azure es menos costoso que *AWS* para *Windows Server* y *SQL Server*.

c) Google Cloud Platform

"*Google Cloud* es un software libre y los entornos de nube híbrida y multinube. Ahora podrás usar tus datos y ejecutar tus aplicaciones en cualquier tipo de nube o entorno. Nuestras soluciones en la nube abierta permiten mantener la coherencia

entre nubes públicas y privadas. Así, las empresas pueden modernizarse y los desarrolladores pueden crear a mayor velocidad, independientemente del entorno" (Cloud, 2021).

Google Cloud protege los datos, las aplicaciones, la infraestructura y los clientes de actividades fraudulentas, correo no deseado y abuso mediante la misma infraestructura y servicios de seguridad que utiliza *Google*. Los servicios de almacenamiento de datos proporcionan almacenamiento, transporte y uso de datos cifrados.

En la Tabla 4, pueden visualizarse las ventajas y desventajas de los diferentes proveedores de servicios en la nube antes descritos.

Tabla 4. Análisis comparativo entre los diferentes proveedores de servicios en la nube.

AWS	***Microsoft Azure***	***Google Cloud Platform***
(AWS, 2021): • 12 meses gratis: Está disponible exclusivamente para los nuevos clientes de *AWS* durante doce meses a partir de la fecha de inscripción en *AWS*. • Pruebas: A corto plazo gratis que comienzan desde el momento en que comienza el primer uso. • Es escalable: Cada usuario puede contratar lo que quiera y configurar su servicio. • Seguridad: Localización de datos, protección y confidencialidad con servicios y funciones integrales.	(Azure, 2021): • 30 días de crédito: Está disponible un crédito por 200 usd después del periodo se cobra por el uso. • Pruebas: Se paga solo si usa más de los importes mensuales gratuitos. • Es escalable: Cada usuario controla los servicios que desea pero tienen costo. • Seguridad: Tiene un modelo de seguridad estándar para detectar, evaluar, diagnosticar y estabilizar.	(Cloud, 2021): • 90 días de crédito: Está disponible un crédito por 300 usd después del periodo se cobra por el uso. • Pruebas: Se paga si se usa más de las pruebas gratuitas al mes. • Es escalable: Cada usuario configura los servicios que requiera pero tienen costo. • Seguridad: Abstracción de problemas de mantenimiento, soporte e instalación.

CAPÍTULO III. MARCO METODOLÓGICO

El término metodología tiene varios significados. En primera instancia, se refiere a un método, ruta o camino para lograr un fin. En términos más formales, éstas “consisten en un sistema de métodos que utilizados en conjunto buscan obtener un resultado durante la ejecución de un proyecto (Lopez, 2021).

Las metodologías que se encuentran descritas en este apartado se refieren a aquellas dedicadas a la gestión de proyectos del área de ciencia de datos, lo cual ofrece la posibilidad de aplicar mejores prácticas, mejorar la comunicación, estandarizar las actividades a realizar y proporcionar herramientas para la ejecución del proyecto y para la toma de decisiones.

Con el propósito de generar una metodología enfocada al preprocesamiento de leyes ambientales mexicanas, este trabajo propone una metodología híbrida que aprovecha determinados elementos de Proceso Estándar entre Industrias para la Minería de Datos *(Cross Industry Standard Process for Data Mining, CRISP-DM)*, Proceso de Ciencia de Datos en Equipo *(Team Data Science Process, TDSP)* y la guía experimental de McGeoch para obtener un corpus estructurado y listo para ser analizado mediante técnicas de procesamiento del lenguaje natural y diseño experimental.

En principio “*CRISP-DM* (*Cross Industry Standard Process for Data Mining*), se ha convertido en un modelo de proceso estándar entre diversas empresas que aplican minería de datos a sus procesos de negocio. Esta metodología consta de 6 fases“ (Figura 4) (Chapman et al., 2000).

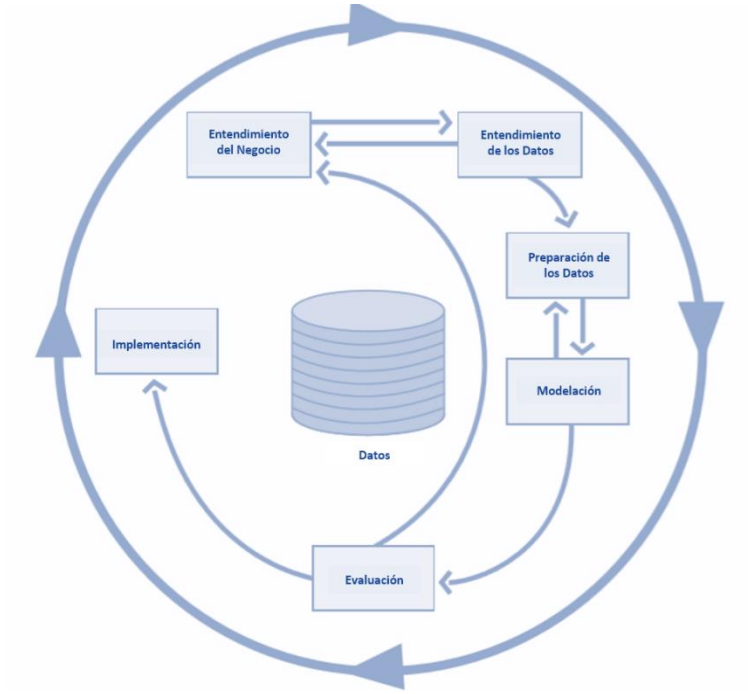


Figura 4. CRISP-DM.

La metodología de "*TDSP* (*Team Data Science Process)*, por su parte, por su carácter ágil e iterativo permite ejecutar y entregar soluciones de analítica avanzada y *machine learning*. Ésta fue diseñada por *Microsoft,* sugiriendo la creación de un repositorio independiente para cada proyecto, con el fin de promover el control de versiones, la seguridad de la información y el trabajo colaborativo. A su vez, proporciona plantillas y carpetas que permiten estructurar los documentos generados dentro de ubicaciones estándar. Esta estructura de carpetas organiza los archivos de tal modo, que es más fácil localizar y entender las entradas y salidas del sistema, así como los códigos utilizados para su procesamiento y los informes de éstos" (Microsoft, 2021).

Finalmente “la metodología propuesta por McGeoch define cómo llevar a cabo un proceso experimental dividido en dos grandes fases, la planeación y la ejecución. Estas fases a su vez poseen tareas específicas para realizar la experimentación a lo largo del proyecto” (McGeoch, 2012).

A continuación, se describen las generalidades de cada una de las metodologías antes mencionadas, así como el proceso de integración de estas en una metodología híbrida.

3.1 METODOLOGÍA CRISP-DM

Con el fin de orientar las actividades a realizar dentro del presente proyecto se propone *CRISP-DM*, el cual es un modelo de proceso de minería de datos que describe la manera en la que los expertos en esta materia abordan el desafio de obtener conocimiento a partir de los datos (Chapman et al., 2000).

3.1.1 Historia de CRISP-DM

“CRISP-DM fue concebido a finales de 1996 en el prematuro mercado de la minería de datos. Daimler Chrysler (entonces Daimler-Benz) estuvo a la cabeza de la mayoría de las organizaciones industriales y comerciales en la aplicación de la minería de datos en sus operaciones comerciales” (Chapman et al., 2000).

La delimitación de las fases y la terminología a utilizar sirvieron de base para probar proyectos de minería de datos a gran escala en una variedad de negocios, lo que contribuyó a brindar experiencia práctica y del mundo real sobre cómo se llevan a cabo los proyectos de minería de datos.

3.1.2 Proceso jerárquico

CRISP-DM se describe en términos de un modelo de proceso jerárquico, que consta de conjuntos de tareas descritas en los siguientes cuatro niveles de abstracción (de lo general a lo específico): fase, tarea genérica, tarea especializada e instancia de proceso. En el nivel superior, el proceso de minería de datos se organiza en varias fases; cada fase consta de varias tareas genéricas y especializadas, que a su vez éstas derivan en instancias de proceso (Figura 5).

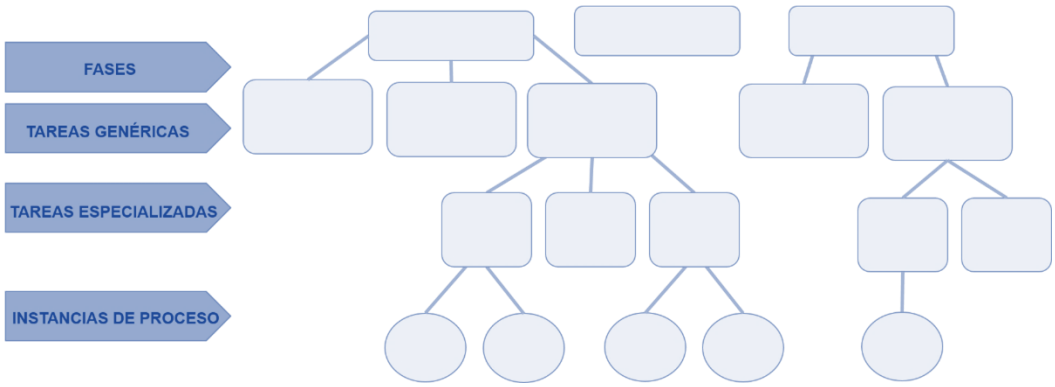


Figura 5. Proceso jerárquico. Fuente: Elaboración propia.

Las fases son los estados sucesivos por los que transcurre el proceso de minería de datos, que serán explicitados más adelante.

Las tareas genéricas forman parte del segundo nivel, que se denomina así porque pretenden ser lo suficientemente amplias, completas y estables. Se consideran amplias, por incluir todos los datos que pueden ser necesarios para la minería de datos; completas, porque incluyen todos los procesos de minería de datos, así como sus aplicaciones; y estables, en el sentido de que el modelo debe ser aplicable a desarrollos futuros.

En el tercer nivel, se incluyen las tareas especializadas, las cuales describen cómo se deben llevar a cabo las acciones de cada tarea general en situaciones específicas. Por ejemplo, en el segundo nivel, puede haber una tarea genérica llamada "limpieza de datos", y en el tercer nivel, se describe cómo varía esta tarea en diferentes situaciones, como la limpieza de valores numéricos frente a la limpieza de valores categóricos, o cómo cambia el método de limpieza según el modelo utilizado.

El cuarto nivel, conocido como instancia de proceso, se refiere al registro de acciones, decisiones y resultados de una operación de minería de datos en tiempo real. Una instancia de proceso se organiza de acuerdo con las tareas definidas en los niveles superiores, pero representa lo que realmente sucedió en un caso específico en lugar de lo que sucede en el esquema general.

Para lograr un buen uso de esta metodología, se procederá a explicar cada una de las fases, tareas y procesos a utilizar en este proyecto. Por tal motivo, sólo se implementan las tres primeras fases, Entendimiento del Negocio, Entendimiento de los Datos y Preparación de los Datos, profundizando en ciertas tareas genéricas y específicas de la tercera fase.

3.2 METODOLOGÍA TDSP

Como complemento a la metodología antes descrita, para la parte estructural de los procesos, se utilizará una segunda metodología denominada *TDSP*, la cual nos ofrece fases similares a las de *CRISP-DM*, pero con apoyo de plantillas que permiten la emisión de resultados apegados a estándares.

3.2.1 Características de TDSP

"El Proceso de Ciencia de Datos en Equipo (*TDSP)*, ayuda a mejorar la colaboración y el aprendizaje en equipo al sugerir cómo los roles de equipo funcionan mejor juntos. Incluye además, procedimientos recomendados y estructuras propuestas por *Microsoft* y otros líderes del sector para ayudar a implementar correctamente iniciativas de ciencia de datos" el cual tiene los siguientes componentes principales (Microsoft, 2021):

- Una definición de ciclo de vida de ciencia de datos.
- Una estructura de proyecto estandarizada.
- Infraestructura y recursos recomendados para proyectos de ciencia de datos.
- Herramientas y utilidades recomendadas para la ejecución de proyectos.

3.2.2 Ciclo de vida de TDSP

TDSP proporciona un ciclo de vida para estructurar el desarrollo de los proyectos de ciencia de datos. (Microsoft, 2021), éste describe las 4 fases principales (Figura 6) por las que pasan normalmente los proyectos, a menudo de forma iterativa:

- Entendimiento del negocio.
- Adquisición y entendimiento de los datos.
- Modelación.
- Implementación.

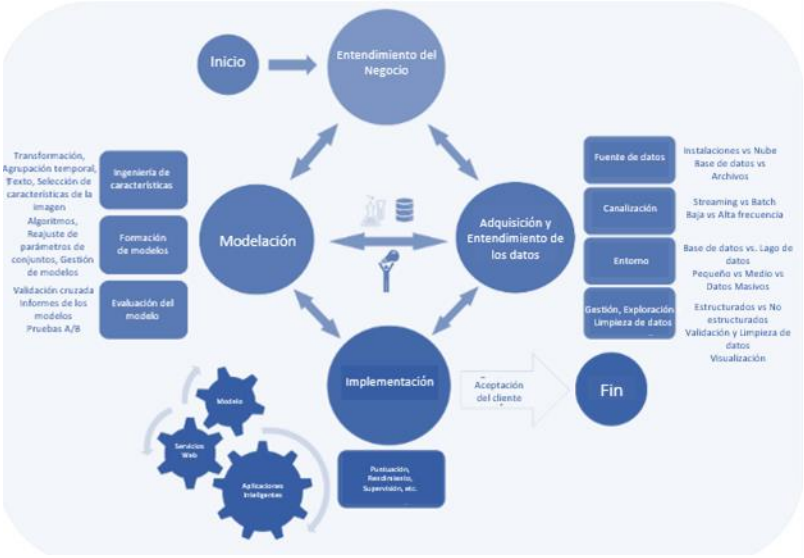


Figura 6. Data Science Lifecycle (Microsoft, 2021).

La metodología híbrida propuesta aprovecha elementos tanto de *CRISP-DM* como de *TDSP*:

- Una estructura de directorios con nombres predefinidos y ubicaciones estándar; sin embargo, se crearon adicionalmente un directorio llamado experimentos que incluye los *scripts* de configuración y otro para el informe de resultados experimentales.
- Definición de los entregables de cada fase: se contemplan 5 para el entendimiento del negocio, 2 para el entendimiento de los datos y 2 para la preparación de los mismos.
- Descripción de la colaboración y el trabajo en equipo. En este sentido, este proyecto a pesar de ser un trabajo de tesis individual pretende cubrir varios roles con la consideración de colaboradores o asesores, es decir, expertos en políticas, especialistas en medio ambiente y en ciencia de datos, entre otros.

Por último, este proyecto incluye una fase relacionada con la validación de la fase de preparación de datos mediante un proceso experimental con el objetivo de planificar y ejecutar sistemáticamente estrategias de prueba.

3.2.3 Esquema de carpetas

Derivado de las características antes descritas, cuando todas las tareas de un proyecto comparten una misma estructura de directorio y usan plantillas para esquematizar los documentos empleados, resulta fácil para los miembros del equipo encontrar la información relativa a cada una de estas tareas.

La estructura de carpetas propuesta por *TDSP* se muestra a continuación en la Figura 7.

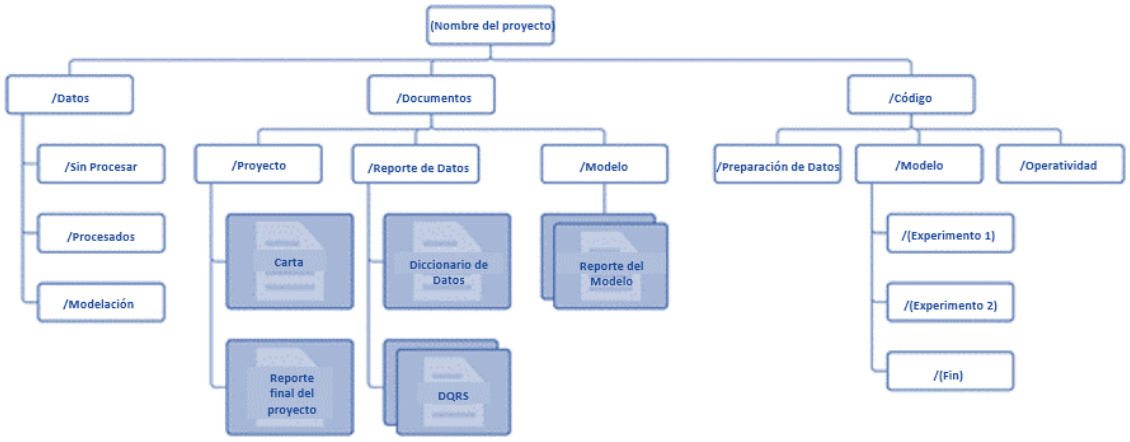


Figura 7. Esquema de carpetas y plantillas de TDSP (Microsoft, 2021).

En la estructura de carpetas y plantillas, se modificaron directorios que incluyen *scripts* de configuración e informes de resultados experimentales (Véase Figura 8).

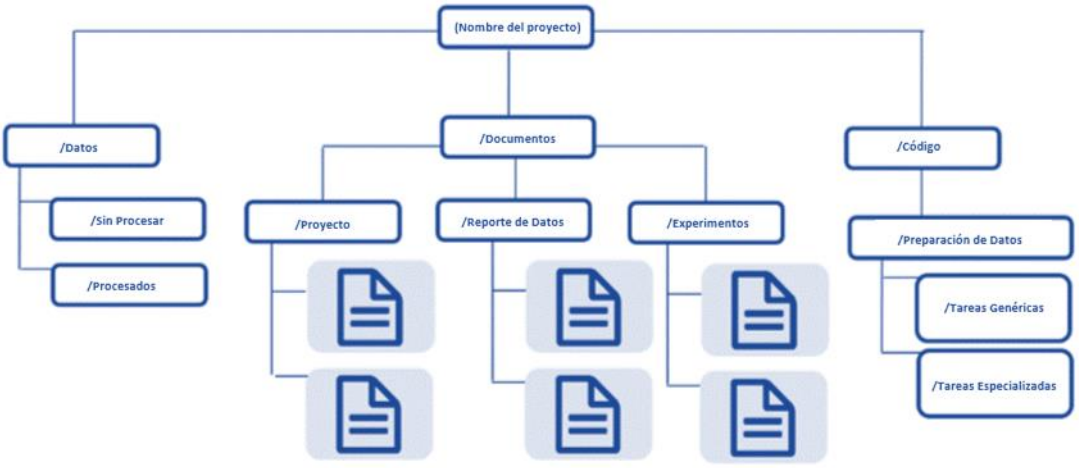


Figura 8. Esquema modificado. Fuente: Elaboración propia.

Finalmente, con el fin de describir la importancia que tiene el realizar experimentos en el área de las ciencias computacionales, se explica una tercera metodología a considerar.

3.3 GUÍA PARA REALIZAR EXPERIMENTOS ALGORÍTMICOS

Antes de realizar experimentos se debe cuestionar ¿Cuál es el objetivo de realizarlos?, ¿Qué se debe medir?, ¿Cómo se analizarán los datos?, entre otras muchas preguntas.

Las respuestas a estas preguntas deberán dar como resultado el diseño de un conjunto de casos de prueba que permitan el análisis de diferentes tareas o algoritmos, lo cual se especificará en los apartados siguientes.

Según Catherine C. McGeoch (2012), el proceso experimental está dividido en dos grandes fases, la planeación y la ejecución:

1 Planificación: Consiste en la formulación de preguntas y se alterna con la creación de herramientas de prueba y el diseño de experimentos.
2 Ejecución: La realización de experimentos se alterna con el análisis de datos. Los pasos individuales pueden llevarse a cabo en diferente orden y, a veces, se omiten.

Estas fases a su vez poseen tareas específicas que se aprecian en la Figura 9:

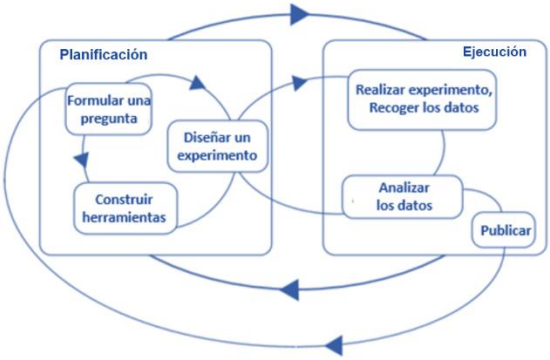


Figura 9. Proceso experimental.

Las tareas que componen cada una de las fases mostradas anteriormente son las siguientes (McGeoch, 2012):

Durante la fase de planificación del experimento:

a. Formular una pregunta. Se formulan preguntas que respondan al experimento que se lleva a cabo.

b. Construir el mejor entorno y/o herramientas. El entorno de prueba incluye el programa de prueba, las instancias de entrada y los generadores de instancias, las herramientas y los paquetes de medición y el software de

análisis de datos. Estos componentes pueden estar fácilmente disponibles o requerir un tiempo de desarrollo considerado en sí mismo.

c. Diseñar un experimento para responder a la pregunta. Especificar, por ejemplo, qué propiedades se van a medir, qué categorías de entrada se aplican, qué tamaños de entrada se miden, cuántos ensayos aleatorios se realizan, etc.

Ahora bien, durante la fase de ejecución del experimento:

a. Ejecutar las pruebas y recopilar los datos. Se ejecutan los experimentos diseñados en la fase anterior.

b. Aplicar el análisis de datos para obtener información y conocimientos. Si la pregunta no ha sido respondida, se debe volver a la fase de planificación e intentarlo nuevamente.

c. Publicar los resultados. En el mejor de los casos la publicación suscita a varias preguntas nuevas, que vuelven a iniciar el proceso.

La naturaleza de los experimentos tiende a evolucionar a medida que avanza el proyecto, por lo cual algunas de las particularidades del proceso que propone McGeoch son las siguientes:

a) Una sola ronda de planificación y experimentación puede formar parte de un proceso más amplio en un proyecto desarrollado con fines de diseño o análisis, o ambos.
b) Los experimentos se llevan a cabo en ciclos dentro de ciclos.
c) La planificación de experimentos se alterna con su ejecución.

d) Los tres pasos de la fase de planificación pueden llevarse a cabo en cualquier orden y los pasos pueden saltarse ocasionalmente.

A continuación, se detalla cómo se interrelacionan las metodologías antes descritas, para crear una metodología híbrida.

3.4 METODOLOGÍA HÍBRIDA

Derivado de la descripción de las dos metodologías utilizadas en proyectos de ciencia de datos y de la guía de diseño experimental, se plantea el uso de una metodología híbrida la cual se explica a través de la Figura 10.

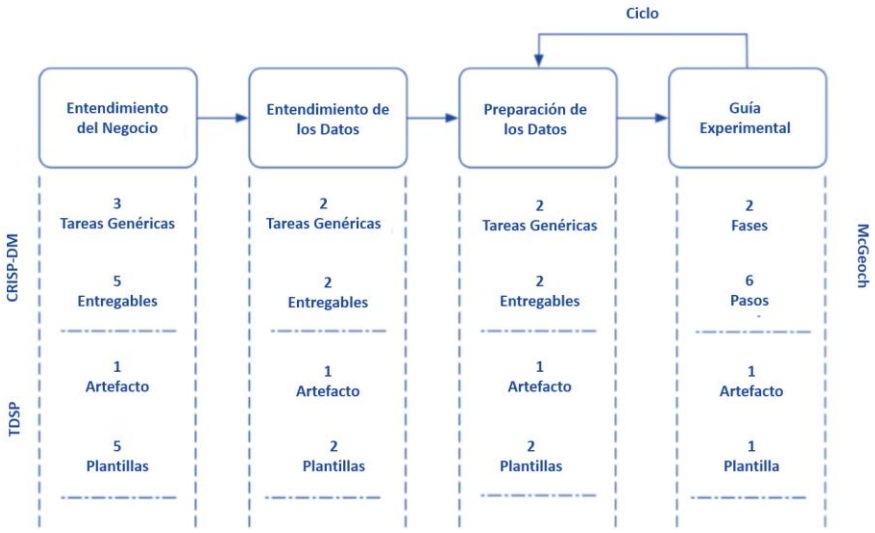


Figura 10. Esquema de interacción entre las metodologías McGeoch, CRISP-DM y TDSP. Fuente: Elaboración propia

En la figura anterior, se muestra una metodología compuesta por las tres fases de *CRISP-DM* a utilizar en este proyecto, junto con las tareas genéricas y entregables de cada fase. *TDSP* añade un artefacto general representado por la estructura de directorios, así como un número variable de plantillas para cada tarea. Por último, la guía experimental mantiene un ciclo de vida activo de planeación y ejecución de las fases y sus respectivas tareas genéricas, controlando la repetición de estas o la adición o modificación de tareas especializadas.

La integración de las metodologías propuestas permite complementar las tareas sugeridas por *CRISP-DM*, las cuales se describirán de acuerdo con las necesidades específicas del proyecto.

3.4.1 Entendimiento del Negocio

La primera fase del proceso, se centra en la comprensión de los objetivos y requisitos del proyecto, convierte este conocimiento en una definición del problema de minería de datos y un plan preliminar diseñado para lograr los objetivos (Figura 11).

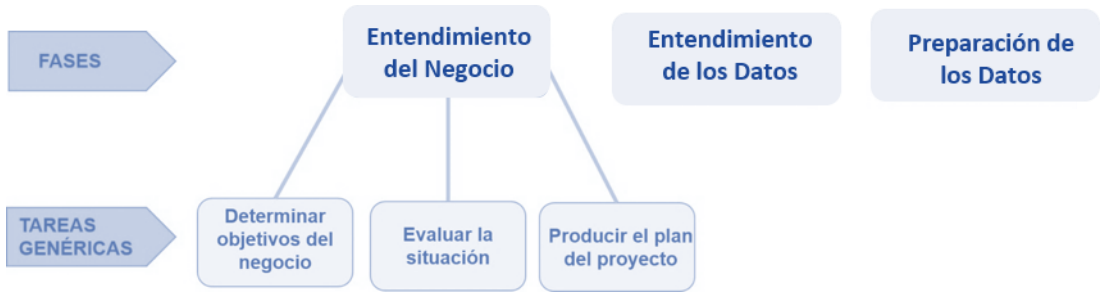


Figura 11. Fase 1 de la Metodología CRISP-DM. Fuente: Elaboración propia.

A continuación, se mencionan y describen brevemente en cada una de sus tareas genéricas. En el Anexo 1 encontrará información más detallada acerca de los entregables obtenidos, la ubicación de carpeta dentro del repositorio *Github* y el llenado de información de proyecto para cada entregable.

a) Determinar los objetivos del negocio

Comprende a fondo desde una perspectiva lo que realmente se quiere conseguir. A menudo se tienen muchos objetivos y limitaciones que compiten entre sí y que deben equilibrarse adecuadamente. Se debe descubrir desde el principio los factores importantes que pueden influir en el resultado del proyecto. Una posible consecuencia de descuidar este paso es perder una gran cantidad de esfuerzo para dar las respuestas correctas a las preguntas equivocadas.

b) Evaluación de la situación

Implica una investigación más detallada de todos los recursos, limitaciones, suposiciones y otros factores que deben tenerse en cuenta para determinar el objetivo del análisis de datos y el plan del proyecto. Otros factores que deben tenerse en cuenta para determinar el objetivo del análisis de datos y el plan del proyecto.

c) Realizar el plan del proyecto

Describe el plan previsto para alcanzar los objetivos de la minería de datos y lograrlos, se deben especificar los pasos a seguir durante el resto del proyecto.

3.4.2 Entendimiento de los Datos

La segunda fase comienza con la recolección inicial de datos y continúa con actividades que permiten entender los datos, identificar problemas de calidad de los datos, descubrir las primeras percepciones de los datos y/o detectar subconjuntos interesantes para formar hipótesis sobre información oculta (Figura 12).

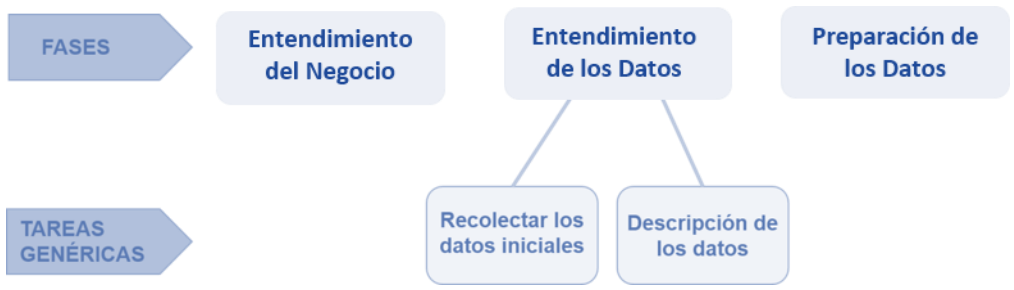


Figura 12. Fase 2 de la Metodología CRISP-DM. Fuente: Elaboración propia.

A continuación, se mencionan y describen brevemente en cada una de sus tareas genéricas. En el Anexo 2 encontrará información más detallada acerca de los entregables obtenidos, la ubicación de carpeta dentro del repositorio *Github* y el llenado de información del proyecto para cada entregable.

a) Recolectar los datos iniciales

Adquirir los datos (o acceder a ellos) enumerados en los recursos del proyecto. Esta recopilación inicial incluye la carga de datos, si es necesario para el entendimiento de los datos.

b) Descripción de los datos

Examina las propiedades "brutas" o "sin procesar" de los datos adquiridos e informar de los resultados.

3.4.3 Preparación de los Datos

La última fase abarca todas las actividades necesarias para construir el conjunto de datos final a partir de los datos iniciales. Es probable que esta tarea se realice varias veces y no en un orden preestablecido. Las tareas incluyen la selección, así como la transformación y limpieza de los datos (Figura 13).

Figura 13. Fase 3 de la Metodología CRISP-DM. Fuente: Elaboración propia.

a) Limpiar los datos

Eleva la calidad de los datos al nivel requerido por las técnicas de análisis seleccionadas. Esto puede implicar la selección de subconjuntos limpios de los datos, la inserción de valores por defecto adecuados.

b) Formateo de los datos

Consiste principalmente en realizar transformaciones sintácticas de datos sin cambiar su significado, de tal manera que facilite el uso de una técnica específica de minería de datos.

En el Anexo 3 encontrará información más detallada acerca de los entregables obtenidos, la ubicación de carpeta dentro del repositorio *Github* y el llenado de información del proyecto para cada entregable.

A partir del seguimiento de la metodología híbrida sugerida, se generan un conjunto de tareas a realizar, las cuales estarán identificadas por las siglas de la fase a la que pertenecen, una G si se trata de una tarea Genérica o una E si es

específica, según la clasificación que propone la metodología *CRISP-DM*, más un número secuencial para identificar el orden de su implementación.

Existen otras fases dentro de la metodología *CRISP-DM*, que son el modelado, la evaluación y el despliegue, pero éstas no se encuentran dentro del alcance del proyecto.

Además de las metodologías descritas, es fundamental especificar la técnica estadística utilizada para la realización de los diversos experimentos, mismos que se muestran en el capítulo siguiente.

CAPÍTULO IV. RESULTADOS DEL DISEÑO EXPERIMENTAL

Dada la metodología híbrida descrita en el capítulo anterior, a continuación, se procede a llevar a cabo la planeación y ejecución de los experimentos que conforman este proyecto, mismos que sólo se aplican a la tercera fase de *CRISP-DM* denominada Preparación de los Datos.

4.1 PLANEACIÓN DEL DISEÑO EXPERIMENTAL

Para el caso de este proyecto, las leyes medioambientales son ejemplo de documentos de texto de un dominio especializado, cuyo proceso de limpieza no está completamente descrito por *CRISP-DM* o *TDSP* y, por lo tanto, se comienza seleccionando las tareas genéricas a realizar y deben personalizarse las tareas especializadas a implementar.

Asimismo, los experimentos comienzan ejecutándose en una infraestructura de equipo de cómputo local y se planea migrar dichos experimentos a la nube. El almacenamiento de la información considera tanto los datos en formato original o crudo (formato pdf) como los preprocesados, en texto plano y en formato json, el código utilizado para su procesamiento y la documentación respectiva.

Derivado de la utilización de *TDSP*, para los experimentos antes mencionados, se utiliza una estructura de carpetas necesarias para la documentación del proyecto. Las carpetas son artefactos distintos que se asocian a los entregables definidos. Los pasos a seguir para cada experimento serán como se indicó en la sección 3.3.

Respecto al análisis estadístico de los experimentos, se determinó utilizar *R Project* para las cuatro tareas de preprocesamiento de este proyecto. A continuación, se explica la obtención de datos y el análisis estadístico para cada una de las tareas anteriores.

4.1.1 Selección de documentos de texto

Los documentos legislativos de México están formados por leyes generales y federales, reglamentos, normas oficiales, constituciones y leyes estatales, entre otros. Además, México participa en poco más de 70 tratados internacionales enfocados en la Protección del Patrimonio Biocultural. “Dada la cantidad y diversidad de leyes en materia ambiental, éstas requieren ser revisadas y analizadas para detectar posibles incoherencias, deliberadas o no, y así proteger a los ecosistemas frente a riesgos y amenazas. Para este importante reto, se debe recurrir a la minería de textos y al procesamiento de lenguaje natural, áreas que utilizan las técnicas y los algoritmos de la inteligencia artificial” (Hidalgo, 2018).

De este conjunto de leyes, se seleccionaron 9 leyes ambientales mexicanas de ámbito federal, las cuales han sido elegidas por un grupo de expertos en política ambiental mexicana, por ser las más relevantes.

Estas se encuentran disponibles en formato pdf y doc desde la página web www.diputados.gob.mx y son de acceso libre. Las 9 leyes ambientales (objeto de estudio en esta tesis) son las siguientes:

1. LAN: Ley de Aguas Nacionales
2. LGEEPA: Ley General del Equilibrio Ecológico y la Protección al Ambiente
3. LFBOGM: Ley Federal de Bioseguridad de Organismos Genéticamente Modificados
4. LDRS: Ley de Desarrollo Rural Sustentable
5. LGCC: Ley General de Cambio Climático
6. LGPAS: Ley General de Pesca y Acuacultura Sustentable

7. LGVS: Ley General de Vida Silvestre
8. LGDFS: Ley General de Desarrollo Forestal Sustentable
9. LGPGIR: Ley General para la Prevención y Gestión Integral de Residuos

Los documentos de ley se encuentran representados principalmente por textos cuyas diferencias radican en el número de páginas, tamaño de los enunciados, uso de diferente tipografía en los párrafos, incorporación de pies de página, encabezados e imágenes, entre otros.

4.1.2 Determinación de herramientas tecnológicas

Para procesar y analizar las leyes, se hace uso de dos lenguajes de programación ampliamente utilizados en el área de ciencia de datos.

En primer lugar, "*Python* que es un lenguaje interpretado de alto nivel, cuya filosofía hace hincapié en la legibilidad de su código. Está desarrollado bajo una licencia de código abierto, lo que lo hace de libre uso y distribución" (Python, 2021).

Se eligieron las bibliotecas PyPDF2 y Pdfplumber debido a que son bibliotecas compatibles con el lenguaje *Python* las cuales son capaces de dividir, fusionar, recortar y transformar el texto que está contenido en el archivo pdf. Además, estas bibliotecas se enfocan en el texto contenido en tablas, figuras, logotipos. Lo anterior no representa un obstáculo para que la biblioteca consiga esta extracción de texto.

En contraste, otras bibliotecas de *Python* llamadas Fitz y PDFMiner no obtuvieron de manera satisfactoria esta extracción de texto. Las bibliotecas específicas utilizadas para la tarea de transformación de formato original pdf a formato de texto plano son:

- PyPDF2 es una biblioteca de *Python* de código abierto y gratuita capaz de agregar datos personalizados, contraseñas a archivos pdf, así como recuperar texto y metadatos (PyPDF2, 2022). Se programaron 55 líneas de código en esta biblioteca para esta tarea.
- Pdfplumber es una biblioteca gratuita de *Python* de código abierto capaz de extraer texto de cualquier página dada. También puede intentar preservar el diseño de un texto, así como identificar las coordenadas de las palabras (Github, 2022). Se programaron 54 líneas de código en esta biblioteca para esta tarea.

Adicionalmente se utiliza el lenguaje de programación *R Project* (versión 4.2.2), orientado al análisis estadístico en el campo de la ciencia de datos. Este software permite realizar procedimientos estadísticos y gráficos, así como modelos lineales y no lineales. Se caracteriza por ser libre y gratuito.

En seguimiento a las fases descritas en la introducción de este capítulo se continúa con la siguiente tarea:

4.1.3 Formulación de preguntas de investigación

a.1 ¿En qué consiste el proceso de transformación de formato original en pdf a formato de texto plano realizado por cada una de las bibliotecas?

a.2 ¿Cuál es el tiempo que toma la extracción de texto de un documento en formato pdf y su copiado/escritura en un archivo en formato de texto plano?

a.3 ¿Qué elementos dificultan la transformación de la tarea de formato original en pdf a formato de texto plano?

A partir de las preguntas anteriores, se identifican las siguientes variables:

4.1.4 Definición de variables

En esta fase se definen las variables de respuesta: el tiempo (milisegundos) y memoria (*bytes*). La variable independiente o factor es el tipo de biblioteca con dos niveles (PyPDF2 y Pdfplumber); la unidad de estudio está representada por los nueve documentos de texto. En la Tabla 5 se detalla los elementos del diseño experimental.

Tabla 5. Elementos del diseño experimental.

Variables de respuesta	Variable independiente	Niveles (Tipo de Biblioteca)	Unidad Experimental
Tiempo (ms)	Tipo de biblioteca	1. PyPDF2	9 documentos
Memoria (bytes)		2. Pdfplumber	

4.1.5 Descripción del experimento

Asimismo, se creó una plantilla como apoyo al diseño experimental. Su elaboración se muestra en el Anexo 4.

Tabla 6. Plantilla completada con los resultados de la biblioteca Pdfplumber.

PLANTILLA DE DISEÑO EXPERIMENTAL								
Nombre Proyecto:	Caracterización de recursos computacionales para la fase de preprocesamiento de mineria de textos							
ID Experimento:	DP_TR_01						**Fecha Programada:**	28/10/2022
							Fecha Experimento:	31/10/2022

Nombre de la tarea:	Transformación de una ley en formato original en pdf a formato de texto plano.		
TIPO DE TAREA			
Fase:	DP = Preparación de los Datos (*Data Preparation*).	**Especializada:**	Transformación pdf a texto plano con cada herramienta seleccionada.
Genérica:	Formateo de datos.		
Nombre del pdf:	LGEEPA: Ley General del Equilibrio Ecológico y la Protección al Ambiente.		
Biblioteca:	Pdfplumber.		
Mediciones:	Tiempo y Memoria.		
Factor y sus Niveles:	2 elevado a la 1 x 9= 18		
Prueba estadística:	Wilcoxon.		
	HIPÓTESIS DE PARTIDA		**VARIABLES**
La variación del tiempo y memoria en la transformación de pdf a txt depende de las bibliotecas utilizadas de *Python*.		**Independientes:**	Tipo de biblioteca en *Python*.
	RESULTADOS ESPERADOS	**Dependientes:**	Tiempo y Memoria.
Obtener un archivo en txt equivalente al original de ley ambiental en pdf.		**Variable(s) Controlada(s):**	Características del equipo de cómputo.

							Variable(s) no controlada(s):	Daños en el equipo de cómputo.			
					DESCRIPCIÓN DEL EXPERIMENTO						
Se medirá el tiempo y memoria utilizados en la transformación de 9 textos (leyes ambientales) en formato pdf y con distinto número de páginas, mediante dos distintas bibliotecas de preprocesamiento de textos en *Python* (PyPDF2 / Pdfplumber). El número de repeticiones en cada biblioteca experimentada se calcula de la siguiente manera: 9 documentos de ley multiplicado por las 2 bibliotecas da un total de 18 unidades experimentales El equipo de cómputo donde se ejecutarán las pruebas es el siguiente: Memoria RAM de 3 GB, disco duro de 500 GB, Procesador AMD Procesador 1.60 Ghz x64 y Windows 8.1 Pro. El preprocesamiento se llevará a cabo sin ejecutar otra tarea en la computadora de manera simultánea.											
					CONCLUSIONES SOBRE EL EXPERIMENTO						
Las versiones utilizadas en la ejecución de la tarea de preprocesamiento (extracción de texto del archivo pdf) son *Python* 3.10, Pdfplumber 0.1.2 y PyPDFf2 2.10.3. Las bibliotecas PyPDF2 y Pdfplumber para la extracción de pdf a txt mostraron ser adecuadas en los resultados. La biblioteca PyPDF2 introduce saltos de página en cada palabra, cuenta guiones; además las palabras no están completas y no reconoce signos de puntuación. En cambio la biblioteca Pdfplumber cuenta mejor las palabras y no los guiones, saltos de página como palabras y reconoce las letras grandes, los pies de página, diferentes puntos, dos puntos, comas, signos de puntuación, acentos, encabezados e imágenes.											

4.2 EJECUCIÓN DEL EXPERIMENTO

Cada documento de ley se procesa con las dos bibliotecas, es decir que cada documento es analizado por PyPDF2 y después es analizado por Pdfplumber. Al ejecutar los experimentos se obtuvieron los siguientes resultados:

"El tiempo utilizado por cada biblioteca corresponde al total de milisegundos ejecutada una sola vez. *Timeit.default_timer()* es un módulo que proporciona una forma sencilla de cronometrar el tiempo para pequeños fragmentos de código" (Python, 2021).

La memoria del proceso corresponde a la memoria virtual consumida, a partir de la capacidad total de 2.94 GB. Para esta medición se utilizó *psutil (python system and process utilities)* que es una biblioteca multiplataforma para recuperar información sobre los procesos en ejecución y la utilización del sistema (CPU, memoria, discos, red, sensores) en *Python*.

Esta biblioteca es útil principalmente para la monitorización del sistema y la gestión de procesos en ejecución. Su sintaxis es *psutil.virtual_memory()* y devuelve estadísticas sobre el uso de memoria del sistema como una tupla con nombre que incluye las siguientes métricas[6].

"Total: La memoria física total (swap exclusivo). Disponible: La memoria que se puede dar instantáneamente a los procesos sin que el sistema entre en *swap*. Esta se calcula sumando diferentes valores de memoria en función de la plataforma y se supone que sirve para monitorizar el uso real de la memoria de forma multiplataforma" (Psutil, 2022).

[6] Expresada en bytes.

4.2.1 Transformación de formato original en pdf a formato de texto plano

Este experimento se realiza a partir de la pregunta de investigación correspondiente, de la implementación de los programas con cada biblioteca y del análisis estadístico de las variables de salida determinadas.

a.1 ¿En qué consiste el proceso de transformación de formato original en pdf a texto plano realizado por cada una de las bibliotecas?

La tarea consiste en convertir los documentos que se encuentran en formato pdf a formato de texto plano (txt). Se dispone de 9 documentos de ley, los cuáles son leyes ambientales mexicanas de ámbito federal.

Para las bibliotecas, el código es el siguiente:

Tabla 7. Códigos de ambas bibliotecas.

PyPDF2	Pdfplumber

#Se selecciona la biblioteca
from PyPDF2 import PdfFileReader, PdfFileWriter
#Se importa la función para medir el tiempo
import timeit
#Se importa la función para medir la memoria
import psutil
#Se inicia el tiempo total
start = timeit.default_timer()
#Se inicia el tiempo del proceso
start1 = timeit.default_timer()
#Se selecciona el documento a transformar
file_path = '.\Ley de Aguas Nacionales.pdf'
#Función para leer el documento de pdf
	#Se selecciona la biblioteca
	import pdfplumber
	#Se importa la función para medir el tiempo
	import timeit
	#Se importa la función para medir la memoria
	import psutil
	#Se inicia el tiempo total
	start = timeit.default_timer()
	#Se inicia el tiempo del proceso
	start1 = timeit.default_timer()
	#Se selecciona el documento a transformar
	file_path = 'Ley de Aguas Nacionales.pdf '
	#Función para leer el documento de pdf

```
pdf = PdfFileReader(file_path)

#Se indica el nombre del archivo en txt que se quiere generar

with open('Ley de Aguas Nacionales.txt', 'w') as f:

#Función para el número de páginas de pdf

    for page_num in range(pdf.numPages):

#Imprimir el número de páginas

        # print('Page: {0}'.format(page_num))

#La función anterior obtienen el número de páginas del pdf

        pageObj = pdf.getPage(page_num)

        try:

#Se convierten esas páginas a txt

            txt = pageObj.extractText()

            print(''.center(100, '-'))

        except:

            pass

        else:

            f.write('Page {0}\n'.format(page_num+1))

            f.write(''.center(100, '-'))

#Finaliza el proceso

            f.write(txt)

    f.close()

#Finaliza el tiempo del proceso

end1 = timeit.default_timer()
```

```
pdf = pdfplumber.open(file_path)

#Se indica el nombre del archivo en txt que se quiere generar

with open('Ley de Aguas Nacionales.txt', 'w') as f:

#Función para el número de páginas de pdf

    for page_num in pdf.pages:

#Imprimir el número de páginas

        # print('Page: {0}'.format(page_num))

#La función anterior obtienen el número de páginas del pdf

        #pageObj = pdf.pages[0]

        try:

#Se convierten esas páginas a txt

            txt = page_num.extract_text()

            print(''.center(100, '-'))

        except:

            pass

        else:

            f.write(''.center(100, '-'))

#Finaliza el proceso

            f.write(txt)

    f.close()

#Finaliza el tiempo del proceso

end1 = timeit.default_timer()
```

4.2.2 Análisis estadístico

Se utilizó la prueba de rangos con Wilcoxon para determinar diferencias entre las bibliotecas PyPDF2 y Pdfplumber con respecto al tiempo utilizado en cada biblioteca y la memoria consumida, además se reportan las medianas con el rango intercuartílico (RIC). Se consideró un nivel de significancia de 0.05. Los datos se representaron mediante gráficos de cajas y alambres. Asimismo, se obtuvieron gráficos de dispersión y el coeficiente de correlación de Spearman para analizar la relación entre el número de páginas con el tiempo y memoria. El análisis estadístico se llevó a cabo utilizando el software *R Project* versión 4.2.2.

"El diagrama de cajas y alambres es una herramienta gráfica que se utiliza para representar una serie de datos numéricos a través de sus cuartiles. De esta manera, se muestran a simple vista la mediana y los cuartiles de los datos, y también pueden representarse sus valores atípicos" (Jmp, 2022).

Los diagramas de dispersión usan una colección de puntos colocados usando coordenadas cartesianas para mostrar valores de dos variables. Al mostrar una variable en cada eje, se puede detectar si existe una relación o correlación entre las dos variables. Se pueden interpretar varios tipos de correlación a través de los patrones mostrados en los diagramas de dispersión. "Estos son: positivo (los valores aumentan juntos), negativo (un valor disminuye a medida que el otro aumenta), nulo (sin correlación), lineal, exponencial y en forma de U. La fuerza de la correlación puede determinarse por la proximidad de los puntos entre sí en el gráfico. Los puntos que terminan muy lejos del conjunto general de puntos se conocen como valores atípicos" (Catalogue, 2022).

La prueba de hipótesis Wilcoxon se utiliza para comparar dos grupos. Se recomienda utilizar cuando los tamaños de muestras son pequeños, además cuando los supuestos de normalidad y homogeneidad de varianzas no se cumplen.

En la Tabla 8 se muestra el diseño de la matriz utilizada para el análisis de datos.

Tabla 8. Matriz de datos.

Ley	Biblioteca	Tiempo (ms)	Memoria (bytes)	Número de páginas
1	1	28.6719	78	128
2	1	37.6744	89.6	112
3	1	11.1584	65.9	73
4	1	14.1313	63.6	71
5	1	15.8024	65	72
6	1	19.6925	92.9	50
7	1	14.2578	87.9	56
8	1	17.4321	74.7	49
9	1	43.5912	94.3	64
1	2	113.4356	89	128
2	2	147.1396	94.5	112
3	2	44.9475	76.4	73
4	2	57.6075	78.8	71
5	2	53.6704	79	72
6	2	61.8336	94.7	50
7	2	123.8179	93.2	56
8	2	58.2579	91.4	49
9	2	47.1765	84	64

a.2 ¿Cuál es el tiempo que toma la extracción de texto de un documento en formato pdf y su copiado/escritura en un archivo en formato de texto plano?

Se observa que la biblioteca PyPDF2 consume menos tiempo en el preprocesamiento de las 9 leyes y el consumo de memoria es menor en la mayoría de los casos respecto a la biblioteca Pdfplumber.

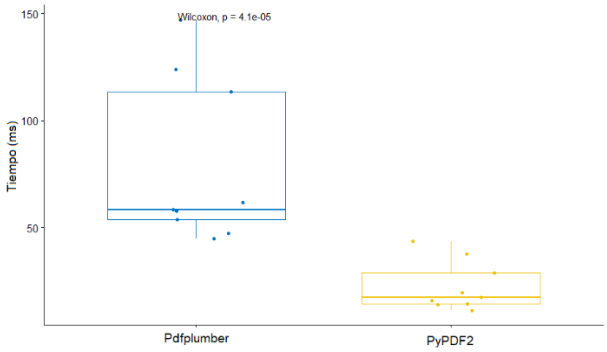


Gráfico 1. Consumo de tiempo para las bibliotecas PyPDF2 y Pdfplumber.

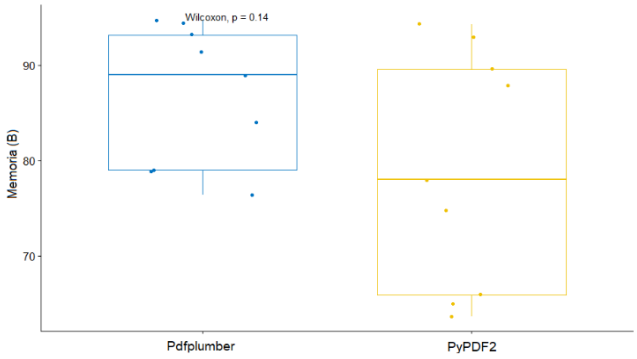


Gráfico 2. Consumo de memoria para las bibliotecas PyPDF2 y Pdfplumber.

En los Gráficos 1 y 2 se puede apreciar que el consumo de tiempo difiere entre las dos bibliotecas (Wilcoxon, W=45, p<0.01), la biblioteca PyPDF2 consume un menor tiempo de preprocesamiento, se obtiene una mediana de 5.43 (RIC: 18.98) ms, en cambio con la biblioteca Pdfplumber el valor de la mediana es mayor 58.26 (RIC: 68.2) ms. Esto se debe a que la biblioteca Pdfplumber presenta una mayor calidad en la transformación de pdf a txt y esto incide en un mayor tiempo.

Respecto al consumo de memoria no se determinaron diferencias (Wilcoxon: W=68, p=0.14), con la biblioteca PyPDF2 la mediana es 78 (RIC:25.80) bytes y con la biblioteca Pdfplumber la mediana resultante fue 89 (RIC: 14.95) bytes. La biblioteca 1 (PyPDF2) consume más memoria que la biblioteca 2 (Pdfplumber) ya que preprocesa los datos en todos los documentos de ley con mejores resultados y calidad en los datos al hacer la transformación a txt.

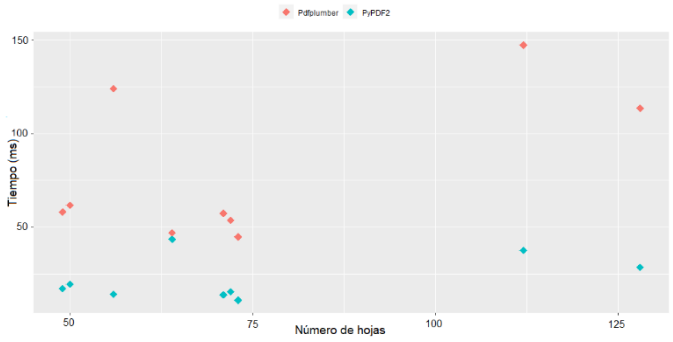


Gráfico 3. Diagrama de dispersión entre el tamaño del documento y el consumo de tiempo en ambas bibliotecas.

En el Gráfico 3 se muestra que no existe relación entre el tamaño del documento y el consumo del tiempo (r=0.341). La mayoría de los documentos varían entre 49 y 73 páginas, solo tres documentos contienen más de cien hojas. Con la biblioteca PyPDF2 los tiempos son menores en comparación con la biblioteca Pdfplumber.

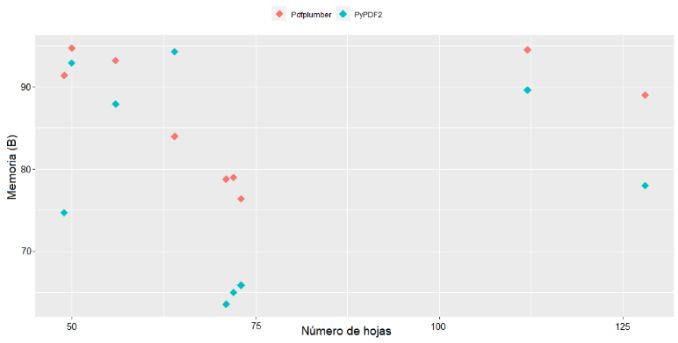


Gráfico 4. Diagrama de dispersión entre el tamaño del documento y el consumo de memoria en ambas bibliotecas.

En el Gráfico 4 se muestra que no existe relación entre el tamaño del documento y el consumo de memoria (r=-0.007). En ambas bibliotecas se muestra que sin importar el tamaño del documento el consumo del tiempo varía entre 60 bytes y 95 bytes. En particular la biblioteca Pdfplumber consume mayor memoria.

4.2.3 Análisis cualitativo

a.3 ¿Qué elementos dificultan la transformación de la tarea de formato original pdf a formato de texto plano?

La biblioteca PyPDF2 tiene saltos de página en cada palabra, cuenta guiones, las palabras no están completas, no reconoce signos de puntuación. En cambio, la biblioteca Pdfplumber tiene un mejor conteo en las palabras porque no las corta en guiones. Esta biblioteca sí reconoce: las letras grandes, los pies de página, diferentes signos de puntuación, por ejemplo: puntos, dos puntos, comas, acentos, encabezados e imágenes.

Tabla 9. Aspectos identificados en las bibliotecas PyPDF2 y Pdfplumber.

PyPDF2	Pdfplumber
En el proceso de transformación consume un menor tiempo y memoria.	El proceso de transformación requiere más tiempo y ocupa mayor memoria, sin embargo, obtiene los resultados esperados.
Examina un menor número de palabras y algunas palabras están incompletas.	Examina un mayor número de palabras y la mayoría de las palabras están completas sin cortes en ellas.
Inserta guiones entre las palabras y los espacios entre párrafos.	Inserta menos guiones entre las palabras y los espacios entre párrafos.
Contiene saltos de página y pies de página.	Contiene pocos saltos de página y pies de página.
Inserta muchos espacios en blanco.	Inserta pocos espacios en blanco.
No reconoce letras de diferente tamaño.	Reconoce letras de diferente tamaño.
No reconoce comas, puntos, signos de puntuación, acentos.	Reconoce comas, puntos, signos de puntuación, acentos.
No reconoce los encabezados.	Reconoce los encabezados y los transforma tal y como se encuentran.
En el texto de una imagen inserta guiones, espacios, etc.	El texto de una imagen se extrae correctamente.

La biblioteca PyPDF2 tiene problemas al transformar ciertos caracteres incluidos en el documento, tales como, letras capitales al inicio de una palabra, fechas, saltos de página y de párrafo, guiones, palabras incompletas y signos de puntuación,

haciendo que el conteo de estas sea mayor. En cambio, la biblioteca Pdfplumber hace un mejor conteo de las palabras, sin alterar el texto cuando se encuentran cualquiera de los elementos antes mencionados, incluyendo las imágenes.

Después de la transformación, se hace una comparación del número de palabras entre el documento transformado a txt y el archivo origen pdf, ya que al transformarse el documento de ley se obtienen adicionalmente, sobre todo para la biblioteca PyPDF2, una serie de palabras cortadas y varios espacios vacíos, que alteran el número de palabras encontradas.

Respecto al tiempo consumido por la biblioteca PyPDF2 notamos que la Ley Federal de Bioseguridad de Organismos Genéticamente Modificados (LFBOGM) obtuvo un 11.1584 ms que fue la que menos tiempo consumió y la ley que mayor tiempo consumió fue la Ley General para la Prevención y Gestión Integral de Residuos (LGPGIR) obteniendo 43.59153 ms. Por su parte, con la biblioteca Pdfplumber la ley que menos tiempo obtuvo fue Ley Federal de Bioseguridad de Organismos Genéticamente Modificados (LFBOGM) con 44.94758 ms y la ley que más tiempo consumió fue la Ley General de Vida Silvestre (LGVS) que obtuvo 123.8179 ms.

Respecto a la memoria consumida por la biblioteca PyPDF2 notamos que la ley que menos memoria consumió fue la Ley de Desarrollo Rural Sustentable (LDRS) obteniendo 63.6 bytes y la ley que más memoria consumió fue la Ley General para la Prevención y Gestión Integral de Residuos (LPGIR) obteniendo 94.3 bytes. Por su parte, con la biblioteca Pdfplumber la ley que menos memoria consumió fue la Ley Federal de Bioseguridad de Organismos Genéticamente Modificados (LFBOGM) obteniendo 76.4 bytes y la que más memoria consumió fue la Ley General de Pesca y Acuacultura Sustentable (LGPAS) que obtuvo 94.7 bytes.

Entre los principales hallazgos, los resultados mostraron que no existe relación entre el tamaño del documento y el consumo del tiempo. Además, en ambas bibliotecas se observó que sin importar el tamaño del documento el consumo

de la memoria varía entre 60 bytes y 95 bytes. En particular la biblioteca Pdfplumber consume mayor memoria pero hace mejor preprocesamiento de estos documentos. Por lo tanto, esta biblioteca se seguirá utilizando para una segunda tarea, la cual consiste en convertir el formato de texto plano a formato json.

Finalmente, se continuará con el análisis cualitativo del conjunto de palabras obtenido por cada biblioteca y los resultados obtenidos servirán de base para la verificación de las tareas de preprocesamiento restantes. Además, será útil para la caracterización de una dupla, tarea y recurso computacional, en una instancia *EC2* en la nube.

Como ejemplo de dicho análisis, se creó un archivo csv con los resultados en forma tabular de la ley de aguas nacionales; en éste se muestra la frecuencia de cada una de las palabras que contiene, el tiempo de ejecución y cantidad de memoria utilizada en bytes (Véase Figura 14).

filename	page_number	num_words	total_time	memory	memory_total	word	frequency
LeydeAguasNacionales.pdf	<Page:112>	55365	323.576512	82.9	2.94	[('LEY',	115
						('DE',	345
						('AGUAS',	115
						('NACIONALES',	114
						('CÁMARA',	112
						('DIPUTADOS',	112
						('DEL',	112
						('H.',	113
						('CONGRESO',	113
						('LA',	113
						('UNIÓN',	112

Figura 14. Archivo CSV de la frecuencia de cada palabra contenida en documentos ambientales

En la Figura 15 se muestran las palabras con menor y mayor frecuencia para así obtener el número de veces que aparece la palabra en el documento, posteriormente se crea un diccionario de palabras mediante un *script* conteniendo *stop words* para así eliminar estas palabras vacías.

filename	page_number	num_words	total_time	memory	memory_total	word	frequency
LeydeAguasNacionales.pdf	<Page:112>	55365	323.5765122	82.9	2.94	('aguas',	341
						('Artículo',	319
						('Secretaría',	236
						('Ley',	233
						('General',	155
						('CÁMARA',	112
						('DIPUTADOS',	112
						('Dávila.-',	1

Figura 15. Palabras con menor y mayor frecuencia.

CAPÍTULO V. CARACTERIZACIÓN DE UNA INSTANCIA EN LA NUBE

Con la finalidad de atender la LGAC de cómputo en la nube se crea la configuración de una máquina virtual y despliegue de una solución tecnológica realizando una o más pruebas escalables en la capa gratuita de *AWS*, utilizando el servicio *EC2*.

5.1 CREACIÓN DE LA INSTANCIA

Para dicha actividad se eligió la Plataforma *Amazon Linux*[7], la cual permite clasificar las instancias mediante el filtro de la capa gratuita,

5.1.1 Selección de la instancia

Por consiguiente, se seleccionan los sistemas operativos de Linux/UNIX.

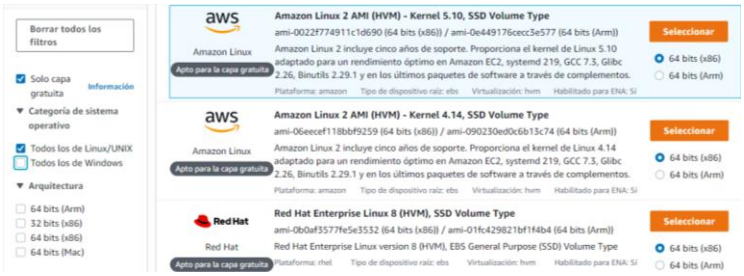


Figura 16. Instancias de la máquina virtual Amazon Linux (Parte 1).

[7] https://us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstances

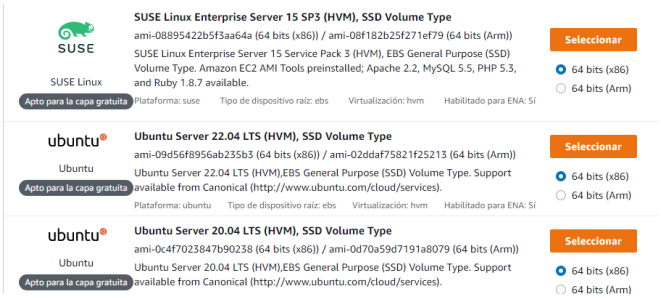


Figura 17. Instancias de la máquina virtual Amazon Linux (Parte 2).

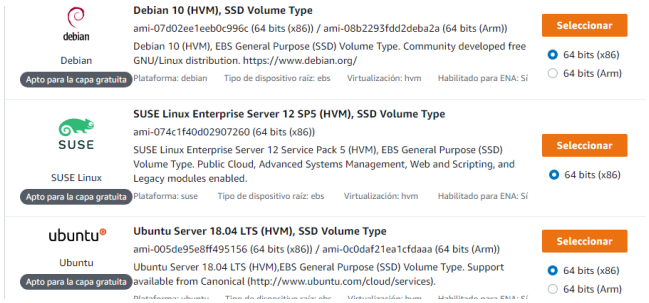


Figura 18. Instancias de la máquina virtual Amazon Linux (Parte 3).

5.1.2 Comparativa de instancias

A continuación, se hace una comparativa entre estas diferentes instancias (Tabla 10).

Tabla 10. Comparación de instancias.

AWS	RedHat	SUSE	Ubuntu	Debian
Amazon Machine Image (AMI) **amzn2-ami-kernel-5.10-hvm-2.0**	RedHad Enterprise Linux 8 (HVM)	SUSE Linux Enterprise Server 15 SP3 (HVM, 64-bit, SSD-Backed)	Ubuntu Server 22.04 LTS (HVM)	Debian 10 (HVM)
Arquitectura: x86_64	Arquitectura: x86_64	Arquitectura: x86_64	Arquitectura: x86_64	Arquitectura: x86_64
Tipo de instancia: t2	Tipo de instancia: t2	Tipo de instancia: t2	Tipo de instancia: t2	Tipo de instancia: t2
Memoria: 8 gb	Memoria:10 gb	Memoria: 10 gb	Memoria: 8 gb	Memoria: 8 gb

Como se muestra la mayoría de las instancias tienen características similares:

- *Amazon Linux 2* incluye cinco años de soporte.
- Proporciona el *kernel* de *Linux 5.10.*
- Adaptado para un rendimiento óptimo en *Amazon EC2.*

La capa gratuita indica que las instancias más adecuadas para el proyecto son las 2 instancias gratuitas en *AWS,* debido a su capacidad de memoria y procesamiento gratuito.

La Tabla 11 muestra que *AWS 5.10* es la instancia adecuada, ya que contiene mayor *kernel*, el cual es el núcleo del sistema operativo y, por tanto, la interfaz entre el *software* y el *hardware* es mejor comparado al *AWS 4.14*.

Tabla 11. Comparación entre instancias AWS.

AWS 5.10	**AWS 4.14**
Amazon Linux 2 Kernel 5.10 AMI 2.0.20220426.0 x86_64 HVM gp2	Amazon Linux 2 AMI 2.0.20220426.0 x86_64 HVM gp2
Arquitectura: x86_64	Arquitectura: x86_64
Tipo de instancia: t2	Tipo de instancia: t2
Memoria: 8 gb	Memoria:8 gb

En *AWS*, el primer año incluye 750 horas de uso de instancias t2.micro (o t3.micro en las regiones en las que t2.micro no esté disponible) en las AMI del nivel gratuito al mes, 30 GiB de almacenamiento de EBS, 2 millones de E/S, 1 GB de instantáneas y 100 GB de ancho de banda a Internet.

5.1.3 Características de la instancia

AWS contiene las siguientes características:

- *Amazon Elastic Compute Cloud* (*Amazon EC2*) proporciona capacidad de computación escalable en la nube de *Amazon Web Services* (*AWS*).

- El uso de *Amazon EC2* elimina la necesidad de invertir inicialmente en hardware, de manera que puede desarrollar e implementar aplicaciones en menos tiempo.
- *Amazon EC2* permite escalar hacia arriba o hacia abajo para controlar los cambios en los requisitos con lo que se reduce la necesidad de prever el tráfico.

Las características utilizadas para el monitoreo y la seguridad de una instancia se pueden apreciar en la Figura 19:

- **ID de la instancia:** El ID de la instancia.
- **Estado de la instancia:** La instancia puede encontrarse en uno de los siguientes estados: pendiente, ejecución, detención, detenida, cerrándose o terminada. Si la instancia está en hibernación, se encuentra en el estado detenida.
- **Tipo de instancia:** El tipo de instancia determina la capacidad de *CPU*, la memoria y el almacenamiento de la instancia.
- **Dirección IP pública:** Se asigna una dirección *IP* pública a la instancia desde el grupo de direcciones *IP* públicas de *Amazon*; la dirección *IP* no está asociada a su cuenta. Al detener la instancia, la dirección *IP* pública se desvincula de su instancia, se vuelve a liberar en el grupo y ya no está disponible para su uso.
- **DNS público:** El nombre de *host* público de la instancia, que se resuelve en la dirección *IP* pública o dirección *IP* elástica de la instancia.
- **DNS privado:** El nombre de *host DNS* privado asignado a la instancia.
- **ID de VPC:** El *ID* de la *VPC* en la que se ejecuta la instancia.
- **ID de subred:** El *ID* de la subred en la que se ejecuta la instancia.
- **Rol de IAM:** El rol de *AWS Identity and Access Management* (*IAM*) asociado a la instancia.

- **Dirección IP propiedad del cliente:** Las direcciones *IP* propiedad del cliente proporcionan conectividad local o externa a los recursos.
- **Dirección IP asignada automáticamente:** La dirección *IP* asignada automáticamente a las nuevas interfaces de red de esta subred.

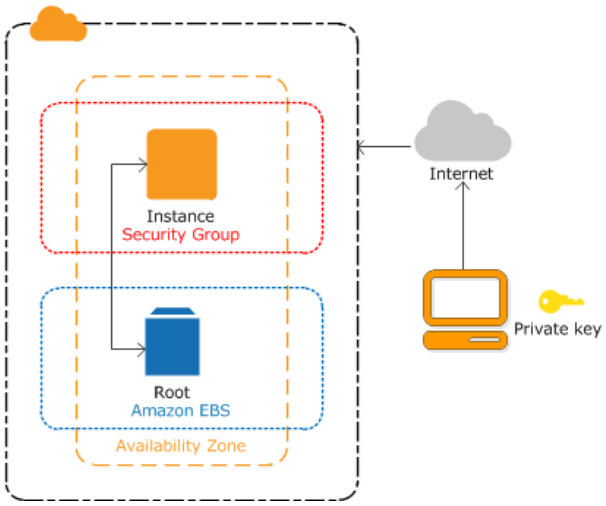


Figura 19. Características de la instancia en AWS.

La configuración de la máquina virtual puede visualizarse en el Anexo 7.

5.2 RESULTADOS DE LA MÁQUINA LOCAL/MÁQUINA VIRTUAL

A continuación, se procede a mostrar los resultados de la ejecución en la máquina local y máquina virtual en cada una de las tareas.

5.2.1 Transformación de formato original en pdf a formato de texto plano

Tabla 12. Resultados en la primera tarea.

Ley	Biblioteca	Tiempo (ms) máquina local	Memoria (bytes) máquina local	Número de páginas	Tiempo (ms) máquina virtual	Memoria (bytes) máquina virtual
1	1	28.6719	78	128	10.6218	74
2	1	37.6744	89.6	112	25.9722	84.4
3	1	11.1584	65.9	73	3.9876	60.3
4	1	14.1313	63.6	71	5.6753	58.2
5	1	15.8024	65	72	8.9076	60
6	1	19.6925	92.9	50	11.8765	85
7	1	14.2578	87.9	56	6.5643	82.4
8	1	17.4321	74.7	49	8.9098	70
9	1	43.5912	94.3	64	24.9099	86
1	2	113.4356	89	128	100.8976	78
2	2	147.1396	94.5	112	127.8976	88.6
3	2	44.9475	76.4	73	31.8765	68
4	2	57.6075	78.8	71	49.9087	70.3

5	2	53.6704	79	72	42.7678	67
6	2	61.8336	94.7	50	52.9876	87.4
7	2	123.8179	93.2	56	112.8763	82.4
8	2	58.2579	91.4	49	44.7650	84.5
9	2	47.1765	84	64	40.7654	70

En los resultados obtenidos por la máquina local se obtuvo un mayor en tiempo y memoria que la máquina virtual como se muestra anteriormente. Debido a la arquitectura de dicha instancia la cual contiene un *kernel* y memoria más potente que la máquina local.

5.2.2 Conversión del formato de texto plano (txt) a formato json

Tabla 13. Resultados en la segunda tarea.

Ley	Biblioteca	Tiempo (ms) máquina local	Memoria (bytes) máquina local	Número de páginas	Tiempo (ms) máquina virtual	Memoria (bytes) máquina virtual
1	1	48.5461	79	128	40.1111	74
2	1	67.7743	89	112	60.1722	84.4
3	1	22.2574	64	73	19.1876	60.3
4	1	28.2314	62	71	20.1753	59
5	1	30.4024	62.5	72	27.1076	58
6	1	40.5928	90.9	50	35.7765	87
7	1	28.3984	85	56	23.4643	82
8	1	35.3221	72	49	30.1098	69.5
9	1	86.3912	93.1	64	80.1011	89.3

1	2	226.2352	89	128	210.2976	84
2	2	227.1194	94	112	217.2976	90
3	2	88.7847	70	73	80.1165	64
4	2	114.7807	79	71	104.1017	75
5	2	104.9870	77	72	98.1178	73
6	2	112.6116	94	50	110.1876	90
7	2	243.4174	96	56	230.1063	92
8	2	116.1379	91	49	106.1450	87
9	2	97.2995	84	64	90.1014	79

De igual forma que en los resultados anteriores, en los resultados obtenidos por la máquina local se obtuvo un mayor tiempo y memoria utilizados que la máquina virtual con comportamiento a la primera tarea, como se muestra anteriormente. Debido a la arquitectura de dicha instancia la cual contiene un *kernel* y memoria más potente que la máquina local.

CAPÍTULO VI. CONCLUSIONES Y TRABAJO FUTURO

6.1 DISCUSIÓN

El preprocesamiento de datos obtiene un conjunto de datos final de calidad y utilidad para la fase de extracción de conocimiento. Sin embargo, dicho preprocesamiento generalmente utiliza la mayor parte del tiempo del tratamiento de los datos, lo que hace conveniente realizar una caracterización de los recursos computacionales empleados para dicha tarea.

Aunado a ello, la generación de una metodología híbrida fue todo un reto, ya que permitió integrar en primera instancia, el Proceso Estándar entre Industrias para la Minería de Datos (*Cross Industry Standard Process for Data Mining, CRISP-DM),* el cual muestra los pasos a considerar para este proyecto en seguimiento a las tareas genéricas y entregables que establece dicha metodología; en segundo lugar, mediante el Proceso de Ciencia de Datos en Equipo *(Team Data Science Process, TDSP)* se añade a cada fase a manera de artefactos, una estructura de directorios y un número variable de plantillas; por último, la guía experimental de McGeoch permitió mantener un ciclo dentro del cual se combinaran las metodologías antes mencionadas, permitiendo la repetición o adición de tareas especializadas de preprocesamiento.

Para comprobar los resultados del proyecto, se realizó la prueba de hipótesis Wilcoxon la cual se utiliza para comparar los tamaños de muestras pequeños, ya que nos referimos a las 9 leyes ambientales en donde los supuestos de normalidad y homogeneidad de varianzas no se cumplieron. Lo que indica que la hipótesis nula donde se refiere que el tiempo y memoria fueran iguales no se cumplió ya que los resultados demuestran que hubo variabilidad en los datos, valiéndose la hipótesis alternativa en donde se demuestra que el tiempo y memoria tuvieron diferentes resultados.

Respecto a la utilización de la instancia *EC2* en modalidad de capa gratuita se debe discutir que después del año de uso, se tienen costos extras que se tienen que cubrir mensualmente, en el caso de la primera tarea denominada transformación pdf a txt el tiempo de uso no fue un problema ya que se situó en este periodo gratuito en la nube, pero al cabo de la segunda tarea denominada conversión txt a json al sobrepasar ese periodo, comenzó a generar costos mensuales sobrepasando el tiempo del año de uso, por lo que se debe realizar una cuenta en *AWS* o migrar el proyecto.

Por último, con respecto a la incompatibilidad de bibliotecas se creía que para la primera tarea genérica todas eran adecuadas, pero las bibliotecas Fitz y PDFMiner fueron incompatibles en el preprocesamiento ya que los datos resultantes no se preprocesaban adecuadamente refiriéndose a que eran saltos de páginas y guiones en estas, asi como al implementarse en la nube generaba errores en sus módulos, lo cual solo fueron compatibles las bibliotecas PyPDF2 y Pdfplumber. En cuanto a la segunda tarea solo se utilizó la biblioteca *Apache Tika* ya que se refiere a una tarea especializada con *scripts* propios para identificar y limpiar las partes de los datos.

6.2 CONCLUSIONES

Ya que el objetivo de este proyecto consistió en medir los recursos computacionales, específicamente tiempo y memoria empleados para realizar tareas de preprocesamiento de documentos de texto, durante la tarea consistente en convertir los documentos que se encuentran en formato pdf a formato de texto plano (txt), se obtuvieron las siguientes conclusiones:

De acuerdo a la prueba estadística se determinaron diferencias significativas en los tiempos indicando que en la biblioteca PyPDF2 consume un menor tiempo con un valor medio de 5.43 ms, este valor medio es de todas las leyes, reportando

una medida de variabilidad de 18.98 ms utilizando el rango intercuartílico, el cual se refiere a la diferencia o variabilidad que hay en los datos.

En cambio, la biblioteca Pdfplumber consume mayor tiempo con un valor medio de 58.26 ms, reportando una medida de variabilidad de 68.2 ms utilizando el rango intercuartílico, el cual se refiere a la diferencia o variabilidad que hay en los datos. Esto se debe a que la biblioteca Pdfplumber presenta una mayor calidad en la transformación de pdf a txt y esto representa un mayor tiempo.

Con respecto al consumo de memoria no se determinaron diferencias con la biblioteca PyPDF2 con un valor medio de 78 bytes obteniendo una medida de variabilidad de 25.80 bytes nuevamente utilizando el rango intercuartílico en el que se ven diferencias en los datos.

En la biblioteca Pdfplumber su valor medio fue 89 bytes, obteniendo una medida de variabilidad de 14.95 bytes viéndose cambios en los datos. En ambas bibliotecas se observó que sin importar el tamaño del documento el consumo de la memoria varía entre 60 bytes y 95 bytes. Esto se debe a que la biblioteca PyPDF2 consume más memoria que la biblioteca Pdfplumber preprocesando los datos en todos los documentos de ley con mejores resultados y calidad en los datos.

Además, se constató que existen elementos que dificultan a la biblioteca PyPDF2 transformar ciertos caracteres incluidos en el documento, tales como, letras capitales al inicio de una palabra, fechas, saltos de página y de párrafo, guiones, palabras incompletas y signos de puntuación, haciendo que el conteo de las mismas sea mayor. En cambio, la biblioteca Pdfplumber hace un mejor conteo de las palabras, sin alterar el texto cuando se encuentra cualquiera de los elementos antes mencionados, incluyendo las imágenes.

Aunado a ello, después de la transformación del archivo, se hace una comparación del número de palabras entre el documento transformado a txt y el archivo origen pdf, ya que al transformarse el documento de ley se obtienen

adicionalmente, sobre todo para la biblioteca PyPDF2, una serie de palabras cortadas y varios espacios vacíos, que alteran el número de palabras encontradas.

Adicionalmente, se utilizaron las bibliotecas Fitz y PDFMiner, las cuales se analizó su eficiencia en la extracción del texto, pero sin medir el tiempo y memoria. Estas sirvieron de base para la elección y comparación de las mejores bibliotecas para esta tarea.

Por último, se visualizó la frecuencia de las palabras que más aparecen en el texto de los documentos, junto con la que menos aparece, por supuesto, la primera corresponde a la preposición “de”, lo que hace necesario realizar un proceso de eliminación de las llamadas *stop words*, y en lo que se refiere a la que menos aparece, corresponde a apellidos de autores, mismos que podrían no ser significativos en la comprensión de una ley. Por tal motivo, se ve la necesidad, de realizar las siguientes tareas de preprocesamiento, que corresponden a conversión de txt a json (incluyendo tokenización), identificación de entidades y etiquetado de textos.

La realización de las tareas de preprocesamiento consiste en la transformación de pdf a text, junto con las pruebas de validación estadística de los resultados obtenidos. Esta tarea sirvió de base para la verificación de las otras tres tareas de preprocesamiento restantes.

6.3 TRABAJO FUTURO

Como trabajos futuros derivados de los experimentos y resultados encontrados, se plantean los siguientes enunciados:

El texto plano que se tiene como resultado de la segunda tarea, se transforma en la forma semiestructurada de los documentos de ley dada por la estructura

jerárquica diseñada. Se preprocesa cada elemento de la estructura usando algoritmos para obtener la parte del discurso (POS), la lematización, el etiquetado y el reconocimiento de entidades nombradas (NER). Se presenta un corpus de leyes mexicanas preprocesadas que puede ser de gran utilidad para el análisis de textos y la ciencia de datos. El uso del lenguaje natural dentro del ámbito jurídico se centra en la comprensión, interpretación, traducción, clasificación, coherencia, búsquedas, etc. Además, se podrían crear aplicaciones para usuarios finales a fin de acceder a esos documentos o para obtener asesoramiento legal. El corpus generado en este proyecto busca, en primera instancia, evaluar el impacto de la legislación mexicana en la implementación de políticas públicas ambientales y de sustentabilidad en el país. Lo cual sería necesario realizar:

- Identificación de entidades. Localización y clasificación de partes del texto estudiado en categorías preestablecidas como lugares, personas, organizaciones, expresiones de tiempo y cantidades. Para realizar el proceso de reconocimiento de entidades se utilizaría spaCy, el cual proporciona una serie de anotaciones lingüísticas sobre la estructura gramatical de un texto, como los tipos de palabras, las partes de la oración, el análisis morfológico, la lematización, el reconocimiento de entidades con nombre y el análisis sintáctico de dependencias. Esta biblioteca funciona con redes neuronales convolucionales y proporciona modelos pre entrenados de distintos idiomas; además, permite crear nuevos modelos o reentrenar los modelos proporcionados con datos propios para crear modelos en campos específicos. Se buscan entidades en el texto y las reemplaza como un solo término. La previa identificación de entidades permitiría que el proceso de etiquetado fuera más eficiente, ya que evitaría la separación en palabras de entidades con nombres compuestos.
- Etiquetado de texto. Clasificación de palabras de un texto (*corpus*) en correspondencia con una parte de la oración determinada, en función de la

definición de la palabra y de su contexto. Consiste en clasificar las palabras de un texto (corpus) en correspondencia con una parte de la oración determinada, en función de la definición de la palabra y de su contexto. El proceso de etiquetado puede llevarse a cabo utilizando la biblioteca spaCy o Freeling la cual es una biblioteca C++ que proporciona funcionalidades de análisis lingüístico, como análisis morfológico, detección de entidades con nombre, etiquetado POS, análisis sintáctico, desambiguación del sentido de las palabras, etiquetado de roles semánticos, etc. El proceso de etiquetado incluye el preprocesamiento de cada elemento de la estructura mediante algoritmos para obtener la lematización, el etiquetado de partes del discurso (POS) y la frecuencia de términos (TF). El etiquetado de partes del discurso (POS) es un proceso de procesamiento del lenguaje natural, que se refiere a la categorización de palabras en un texto (corpus) en correspondencia con una parte particular del discurso, según la definición de la palabra y su contexto.

- Pruebas con diferentes máquinas virtuales. Pruebas con diferentes instancias en la nube para mejorar resultados.
- Pruebas con diferentes sistemas operativos. Pruebas con diferentes arquitecturas de sistema operativo para observar el comportamiento entre estas y elegir los mejores resultados.
- Comportamiento entre máquinas virtuales y sistemas operativos
- Comparación entre las instancias en la nube y los sistemas operativos y elegir la mejor opción.

6.4 PUBLICACIONES

En este proyecto se presentaron tres publicaciones a nivel nacional, las cuales se describen a continuación:

1. Cartel “Técnicas asociadas al preprocesamiento de textos. Caso de estudio: leyes ambientales”, en el marco del 3er Congreso Internacional Multidisciplinario #EsCuestiondeIngenio2021 del Instituto Tecnológico Superior de Xalapa, el día 26 de noviembre del 2021.
2. Ponencia “A Hybrid Methodology based on CRISP-DM and TDSP for the execution of preprocessing tasks in Mexican Environmental Laws”, en el 21st Mexican International Conference on Artificial Intelligence (MICAI 2022) del Tecnológico de Monterrey (ITESM), el día 26 de octubre del 2022.
3. Ponencia “Comparación de los recursos de tiempo y memoria de textos” en la X Jornada de Ciencia y Tecnología Aplicada del Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET/TECNM) en la ciudad de Cuernavaca, Morelos. Dicha ponencia fue el resultado de la estancia en el Laboratorio de Investigación y Asesoría Estadística (LINAE) de la Universidad Veracruzana, el 26 de abril de 2023.

REFERENCIAS BIBLIOGRÁFICAS

Aguilar, M. (2017). El caso del proyecto de la mina La Paila, Municipio de Alto Lucero, Veracruz. In *Gobierno del Estado de Veracruz*.

Aho, A. ., M.S, L., Sethi, R., & Ullman, J. . (2007). *Compilers: Principles, Techniques, and Tools*.

AWS. (2021). *AWS*. 2021. https://aws.amazon.com/es/what-is-aws/

Azure. (2021). *¿Qué es Azure?* 2021. https://azure.microsoft.com/es-mx/overview/what-is-azure/

Barrera, M. C. (2014). Minería de texto: una visión actual. *Biblioteca Universitaria*, *17*(2), 129–138. http://www.redalyc.org/articulo.oa?id=28540279005

Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*.

Catalogue, T. D. V. (2022). *Diagrama de Dispersión*. https://datavizcatalogue.com/ES/metodos/diagrama_de_dispersion.html

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Shearer, C., & Wirth, R. (2000). *CRISP-DM 1.0*.

Cisco. (2021a). *¿Qué es la computación en nube?* 2021. https://www.cisco.com/c/es_mx/solutions/cloud/what-is-cloud-computing.html

Cisco. (2021b). *Tipos de modelos de implementación de computación en nube*. 2021. https://www.cisco.com/c/es_mx/solutions/cloud/what-is-cloud-computing.html#~cloud-computing-deployment-models

Cisco. (2021c). *Tipos de servicios de computación en nube*. 2021. https://www.cisco.com/c/es_mx/solutions/cloud/what-is-cloud-computing.html#~cloud-computing-services

Cloud, G. (2021). *Google Cloud*. 2021. https://cloud.google.com/?utm_source=google&utm_medium=cpc&utm_campaign=latam-MX-all-es-dr-BKWS-all-all-trial-e-dr-1011454-LUAC0010195&utm_content=text-ad-none-any-DEV_c-CRE_512379899411-ADGP_Hybrid %7C BKWS - EXA %7C Txt ~ GCP_General-KWID_43700062784667

CNDH. (2018). *Politicas Públicas*. 2018. https://desca.cndh.org.mx/Politicas/Programas_Nacionales

Conahcyt. (n.d.). *Sistemas Socioecológicos y Sustentabilidad*. https://conacyt.mx/pronaces/pronaces-sistemas-socioecologicos/#:~:text=El Programa Nacional Estratégico en,aprovechamiento de los ecosistemas%2C de

Congreso general de los estados Unidos Mexicanos. (2007). *Ley de Desarrollo Rural Sustentable*. https://www.cmdrs.gob.mx/sites/default/files/cmdrs/sesion/2019/05/15/1801/materiales/4-ldrs-analisiscederssa.pdf

Congreso general de los estados Unidos Mexicanos. (2015). Ley General de Equilibrio Ecológico y Protección al Ambiente. *Semarnat*, 1–128. https://biblioteca.semarnat.gob.mx/janium/Documentos/Ciga/agenda/DOFsr/148.pdf

Congreso general de los estados Unidos Mexicanos. (2018a). Ley General de Pesca y Acuacultura Sustentable. *Diario Oficial de La Federación*, *DOF 24-04-2018*, 63. http://www.diputados.gob.mx/LeyesBiblio/pdf/LGPAS_240418.pdf

Congreso general de los estados Unidos Mexicanos. (2018b). Ley General Vida Silvestre. *Diario Oficial de La Federación*, 1–71. https://www.senado.gob.mx/comisiones/medio_ambiente/docs/LGVS.pdf

Congreso general de los estados Unidos Mexicanos. (2020a). Ley de Aguas Nacionales. *LEY DE AGUAS NACIONALES Nueva*, 91.

Congreso general de los estados Unidos Mexicanos. (2020b). *Ley General de Cambio Climático*. 1–45. http://www.diputados.gob.mx/LeyesBiblio/pdf/16_060120.pdf

Cortez, R. A. (2018). Extracción de conocimiento a partir de textos obtenidos de Twitter. *Entorno*, *65*, 30–41. https://doi.org/10.5377/entorno.v0i65.6048

Cuántico, E. jarocho. (2020). Integralidad Gamma. *La Jornada Veracruz*.

Datos, ciencia de. (2016). *Prueba de los rangos con signo de Wilcoxon le*. https://www.cienciadedatos.net/documentos/18_prueba_de_los_rangos_con_signo_de_wilcoxon

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). Knowledge Discovery in From Data Mining to Databases. *AI Magazine*, *17*(3), 37–54. https://doi.org/10.1007/978-3-319-18032-8_50

Feldman, R., & Sanger, J. (2007). *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*.

Fidencio, J., Lázaro, J., & Reyna-ángeles, E. (2017). *Revista de Cómputo Aplicado automática de tweets en español*. *1*(2), 1–11.

Galvan, J. (2005). *Gaceta del Senado*. 2005. https://www.senado.gob.mx/64/gaceta_del_senado/documento/7622#:~:text=133.,suprema de toda la Unión.

Github. (2022). *Pdfplumber*. https://github.com/jsvine/pdfplumber#python-library

González, F., Ortiz, T., & Sánchez, R. (2020). *Uso responsable de la IA para las políticas públicas: manual de ciencia de datos.* 2020. https://publications.iadb.org/publications/spanish/document/IA-Responsable-Manual-tecnico-Ciclo-de-vida-de-la-inteligencia-artificial.pdf

Guzman Monter, J. L. (2017). Jerarquía del Orden Jurídico Mexicano. In *2017*. https://www.uaeh.edu.mx/docencia/P_Presentaciones/prepa2/2018/Jerarquia_del_Orden_Juridico.pdf

Hernández, J. M. (2016). *Análisis automático de textos en español utilizando NLTK*.

Hidalgo, M. (2018). *Ayudando al medio ambiente con inteligencia artificial*.

Inecol. (2021). *Integralidad Gama*. 2021. http://www.inecol.mx/inecol/index.php/es/component/content/article/17-ciencia-hoy/1398-avisos-de-privacidad-en-proyectos-de-investigacion

Jmp. (2022). *Diagrama de caja*. https://www.jmp.com/es_co/statistics-knowledge-portal/exploratory-data-analysis/box-plot.html

Kao, A., & Poteet, S. R. (2007). *Natural Language Processing and Text Mining*.

Lopez, J. (2021). *Metodologías de Gestión de Proyectos*. 02 Agosto. https://opmintegral.com/gestion-de-proyectos/metodologias-de-gestion-de-proyectos/

Luhn, H. P. (1958). *The automatic creation of literature abstracts*. IBM Journal of Research and Development.

Martinez-Seis, B., Pichardo-Lagunas, O., Koff, H., Equihua, M., Perez-Maqueo, O., & Hernández-Huerta, A. (2022). Unified, Labeled, and Semi-Structured Database of Pre-Processed Mexican Laws. *Data*, *7*(7), 1–13.

https://doi.org/10.3390/data7070091

McGeoch, C. (2012). *A Guide to Experimental Algorithmics* (C. U. Press (ed.)).

Mejía, H. (2000). *Ley general Sobre Medio Ambiente y Recursos Naturales (64-00)*. 1–119.

Mexicanos, C. general de los estados U. (2018). *Ley General De Desarrollo Forestal Sustentable*. 1–70. http://www.ncbi.nlm.nih.gov/pubmed/19302767

Mexicanos, C. general de los estados U. (2020). *Ley de Bioseguridad de Organismos Genéticamente Modificados*. 1–44.

Mexicanos, C. general de los estados U. (2021). *Ley General para la Prevención y Gestión Integral de los Residuos*. 1–156. https://n9.cl/xtw3f

Microsoft. (2021). *¿Qué es el Proceso de ciencia de datos en equipo (TDSP)?* 2021. https://docs.microsoft.com/es-es/azure/architecture/data-science-process/overview

Narave, H., & Cházaro, M. de J. (2017). *La Paila un proyecto ambientalmente inviable: necesidad de fortalecer la legislación de protección ambiental*.

NLTK. (2021). *NLTK*. 2021. https://www.nltk.org/

Palacio, A. (2015). Técnicas de Minería de datos aplicado a la monitorización de sistemas (Data Mining applied to the System Monitoring). In *Universidad de Cantabria*.

Pichardo, O., Martinez, B., Carrera, V., & UPIITA-IPN. (2020). Interrogando Datos en Legislación Ambiental. *La Jornada Veracruz*, 8.

Psutil. (2022). *Psutil documentation*. https://psutil.readthedocs.io/en/latest/

PyPDF2. (2022). *PyPDF2*. https://pypdf2.readthedocs.io/en/3.0.0/

Pypi. (2022). *Tika*. https://pypi.org/project/tika/

Python. (2021). *Python*. 2021. https://www.python.org/

Qingkai, K., Siauw, T., & M. Bayen, A. (2020). *Text Data Mining*.

Sanchez, A. (2021a). *Numpy*. 2021. https://aprendeconalf.es/docencia/python/manual/numpy/

Sanchez, A. (2021b). *Pandas*. 2021. https://aprendeconalf.es/docencia/python/manual/pandas/

Solutions, K. D. (2021). *Diferencia entre datos estructurados y no estructurados*. 2021. https://www.kyoceradocumentsolutions.es/es/smarter-workspaces/insights-hub/articles/diferencia-entre-datos-estructurados-y-no-estructurados.html#:~:text=El 80 %25 de la información,estructurada%2C principalmente en formato texto.

Taeho, J. (2019). Text Mining Concepts, Implementation, and Big Data Challenge. In *Springer* (Vol. 45). https://doi.org/10.1053/j.semdp.2019.02.002

Tandel, S. S., Jamadar, A., & Dudugu, S. (2019). A Survey on Text Mining Techniques. *2019 5th International Conference on Advanced Computing and Communication Systems, ICACCS 2019*, *Icaccs*, 1022–1026. https://doi.org/10.1109/ICACCS.2019.8728547

UPIITA-IPN. (2019). *Primer Coloquio Interdisciplinario para el Análisis de Legislación Ambiental*. 14 de Agosto 2019. https://www.upiita.ipn.mx/novedades/coloquio-legislacion-ambiental

W3schools. (2022). *Json*. https://www.w3schools.com/python/python_json.asp

Zong, C., Xia, R., & Zhang, J. (2021). Text Data Mining. In *Tsinghua University Press by Springer*. https://doi.org/10.1007/3-540-57253-8_65

ANEXO 1. IMPLEMENTACIÓN DE LA FASE DE ENTENDIMIENTO DEL NEGOCIO

Para la ejecución de esta fase, se llevaron a cabo las siguientes tareas genéricas. Para cada una de ellas se describe la nomenclatura, ubicación y entregable generado.

Entregable: BU-G1: Determinar los objetivos del negocio

Ubicación: Doc/Project

https://github.com/YesseniaD/CRCPMT-Project/tree/main/Docs/Project/BU1.%20Objetivos%20del%20Negocio

Tabla 14. Entregable DU-G1-Objetivos del Negocio.

Objetivo:	Medir los recursos computacionales como tiempo y memoria empleados para realizar tareas de preprocesamiento de documentos de texto, específicamente en la fase de preparación de datos mediante técnicas de minería de textos con la finalidad de caracterizar una dupla (una tarea y un recurso computacional) y de este modo, crear una instancia EC2 para aprovechar el cómputo en la nube.
Objetivos específicos:	• Identificar y ejecutar tareas de pre-procesamiento de textos genéricas y especializadas con el fin de homogeneizar los documentos de leyes ambientales mexicanas (generación de un corpus legislativo) utilizando operadores de minería de textos proporcionados por las herramientas y/o lenguajes de programación específicos. • Comprender los objetivos del negocio (business undestanding) y de los datos (data understanding). • Monitorear los recursos (tiempo y memoria) a partir de diseños experimentales, utilizando scripts de *Python* para el

	preprocesamiento de los datos, es decir, 9 documentos legislativos de carácter ambiental seleccionados. • Realizar una o más pruebas escalables con el número de documentos y con las diferentes tareas de preprocesamiento de texto a realizar en la capa gratuita de la máquina virtual de AWS.

Tabla 15. Entregable DU-G1-Criterios de Éxito.

Criterios:	La problemática a resolver mediante este proyecto, radica en aprovechar las funcionalidades que proveen tanto herramientas de software como lenguajes de programación como *Python*, para el análisis de diversos documentos de texto, enfocándose en la fase de preparación de datos. Dichas funcionalidades se aplican a documentos que tratan sobre legislación ambiental en México, los cuales, aunque vienen en formato pdf, internamente presentan una estructura diversa, por lo que su análisis resulta más desafiante. En este sentido, además de ejecutar tareas genéricas y específicas de preparación de los documentos ambientales elegidos, buscamos ofrecer datos cuantitativos acerca de la memoria y el tiempo transcurridos para las distintas configuraciones de tareas de preprocesamiento aplicadas, bajo una infraestructura de hardware determinadas.

Entregable: BU-G2: Evaluar la situación

Ubicación: Docs/Project

https://github.com/YesseniaD/CRCPMT-Project/tree/main/Docs/Project/BU2.%20Evaluar%20la%20Situaci%C3%B3n

Tabla 16. Entregable BU-G2- Inventario de Recursos.

Datos de interés:	La legislación ambiental mexicana presenta formatos y contenidos que no se apegan a una estructura determinada, por lo que su análisis resulta desafiante. Si bien la minería de textos utiliza técnicas y algoritmos para preprocesar documentos, no siempre se está consciente del consumo de recursos computacionales requeridos para realizar una o más tareas específicas.
Recursos:	• Lista de recursos computacionales ○ Inventario (Centro de Cómputo y Computadora Personal) • Recursos humanos

	○ Director del proyecto, codirector del proyecto, investigadoras del IPN con experiencia en PLN • Software ○ *Python* y bibliotecas, Github (Tika, NLTK, PyPDF2 + 2 más) • Listado de restricciones ○ 9 leyes • Recursos computacionales ○ Tiempo y memoria

Tabla 17. Entregable BU-G2- Terminología.

UPIITA-IPN:	Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas del Instituto Politécnico Nacional.
INECOL:	Instituto de Ecología.
PLN:	La minería de textos está relacionada con el Procesamiento del Lenguaje Natural (PLN), que incluye técnicas de inspiración lingüística, es decir, un texto se analiza típicamente desde un punto de vista léxico y sintáctico utilizando una gramática formal, la información resultante se interpreta semánticamente y se utiliza para extraer información sobre lo dicho (Kao & Poteet, 2007).
Preprocesamiento:	Se refiere a la realización de operaciones o transformaciones sobre el texto en algún tipo de representación estructurada o semiestructurada que facilite su posterior análisis. Esta fase se lleva a cabo sobre un conjunto de documentos objetos de estudio determinando el tipo de representación o patrones contenidos en los textos (Taeho, 2019).

Entregable: BU-G3: Producir el plan del proyecto

Ubicación: Docs/Project

https://github.com/YesseniaD/CRCPMT-Project/tree/main/Docs/Project/BU3.%20Plan%20del%20Proyecto

Tabla 18. Entregable BU-G3 - Plan del Proyecto.

Fases/Tareas	Tipo	Acciones	Entregables
BU = Entendimiento del Negocio (Business Understanding).	**Fase**	Entender el negocio	Recopilación de información.
BU-G1 Determinar los objetivos del negocio	Tarea Genérica	Describir el objetivo principal del cliente desde el punto de vista del negocio.	1. Objetivos del negocio. 2. Criterios de éxito.
BU-G2 Evaluar la situación	Tarea Genérica	Hardware, software y recursos humanos.	1. Inventario de recursos. 2. Terminología.
BU-G3 Producir el plan del proyecto	Tarea Genérica	Enumerar los recursos necesarios, las funciones, las responsabilidades y los costes y beneficios. Además de especificar cómo se van a combinar las dos metodologías.	Plan del Proyecto.
BU-E1 Creación del Repositorio de GitHub	Tarea Especializada	Crear un repositorio en *GitHub*.	1. Carpetas. 2. Descripción de las carpetas o repositorios.
DU = Entendimiento de los Datos (Data Understanding).	**Fase**	Entender los datos	Recopilación de datos.
DU-G1 Recolección de Datos Iniciales	Tarea Genérica	Enumerar los documentos junto con su ubicación y el procedimiento para acceder a ellos.	Reporte inicial de recopilación de datos.
DU-G2 Descripción de los datos (Leyes	Tarea Genérica	Describir los datos adquiridos, incluyendo su formato, cantidad e identidad de los	Reporte de descripción de datos.

Ambientales en General)		campos (Estructura de leyes (Logo, nombre, artículos))	
DP = Preparación de los Datos (Data Preparation).	**Fase**	Preparar los datos	Datos listos para ser preparados.
DP-G1 Limpiar datos	Tarea Genérica	Describir las transformaciones a realizar sobre los datos para su limpieza.	Reporte de procesos a aplicar de limpieza de datos.
DP-E1 Transformación de pdf a texto plano	Tarea Especializada	Transformar las leyes de formato pdf a txt mediante las bibliotecas de *Python*.	1. Plantilla de Diseño experimental a realizar. 2. Algoritmo de transformación pdf a txt (dos bibliotecas). 3. Texto de las leyes ambientales transformadas a txt. 4. Tablas de resultados. 5. Análisis e interpretación de resultados.
DP-E2. Eliminación de ruido (Quitar encabezados 1 vez)	Tarea Especializada	Quitar encabezados en todas las páginas, exceptuando en la primera.	Código depurado.
DP-E3. Quitar stopwords, paginas, logos	Tarea Especializada	Quitar stopwords o pies de página a algunas hojas.	Código de limpieza.
DP-G2 Formateo de datos	Tarea Genérica	Describir los cambios sintácticos realizados para satisfacer el objetivo empresarial.	Informe de datos reformulado.
DP-E4. Transformar de TXT a estructura JSON	Tarea Especializada	Transformar txt a json.	Código txt a json.
DP-E5 Descripción del corpus	Tarea Especializada	Describir los textos de leyes ambientales con el fin de servir como muestra representativa para el análisis.	Conjunto cerrado de textos o de datos destinado a la investigación científica.

Entregable: BU-E1: Creación del Repositorio de GitHub

Ubicación: Docs/Project

Tabla 19. Entregable BU-E1 – Repositorio.

Repositorio:	**https://github.com/YesseniaD/CRCPMT-Project**

ANEXO 2. IMPLEMENTACIÓN DE LA FASE DE ENTENDIMIENTO DE LOS DATOS

Para la ejecución de esta fase, se describe la nomenclatura, ubicación y entregable generado de cada una de las tareas genéricas.

Entregable: DU-G1: Recolección de datos iniciales

Ubicación: Data/Raw

https://github.com/YesseniaD/CRCPMT-Project/tree/main/Data/Raw/DU1.%20Recolecci%C3%B3n%20de%20Datos

Tabla 20. Entregable DU-G1- Reporte de adquisición de datos.

Las leyes a analizar son 9 leyes ambientales mexicanas de ámbito federal, las cuales, “establecen las normas para la conservación, protección, mejoramiento y restauración del medio ambiente y los recursos naturales que lo integran” (Mejía, 2000).
Las leyes ambientales seleccionadas para el presente proyecto se encuentran en formato pdf todas disponibles en el sitio web: https://web.diputados.gob.mx/ • LGEEPA: Ley General del Equilibrio Ecológico y la Protección al Ambiente • LAN: Ley de Aguas Nacionales • LDRS: Ley de Desarrollo Rural Sustentable • LGPAS: Ley General de Pesca y Acuacultura Sustentable • LGVS: Ley General de Vida Silvestre • LGDFS: Ley General de Desarrollo Forestal Sustentable • LGPGIR: Ley General Para la Prevención y Gestión Integral de Residuos • LFBOGM: Ley Federal de Bioseguridad de Organismos Genéticamente Modificados • LGCC: Ley General de Cambio Climático

Entregable: DU-G2: Descripción de los datos

Ubicación: Docs/Project

https://github.com/YesseniaD/CRCPMT-Project/tree/main/Docs/Project/DU2.%20Descripci%C3%B3n%20de%20Datos

Tabla 21. Entregable DU-G2 – Reporte de descripción de los datos.

Ley Ambiental	Descripción
LGEEPA: Ley General del Equilibrio Ecológico y la Protección al Ambiente	Preservación y restauración del equilibrio ecológico, así como a la protección al ambiente, en el territorio nacional y las zonas sobre las que la nación ejerce su soberanía y jurisdicción (Congreso general de los estados Unidos Mexicanos, 2015).
LAN: Ley de Aguas Nacionales	En materia de aguas nacionales; es de observancia general en todo el territorio nacional, sus disposiciones son de orden público e interés social y tiene por objeto regular la explotación, uso o aprovechamiento de dichas aguas, su distribución y control, así como la preservación de su cantidad y calidad para lograr su desarrollo integral sustentable (Congreso general de los estados Unidos Mexicanos, 2020a).
LDRS: Ley de Desarrollo Rural Sustentable	Integrar una política de Estado para el desarrollo rural, por encima de las naturales diferencias entre las fuerzas políticas, capaz de construir acuerdos en puntos básicos que garanticen metas y programas en el largo plazo, creadora de seguridad, confianza y certidumbre; como una de las principales aspiraciones de los productores y sus organizaciones (Congreso general de los estados Unidos Mexicanos, 2007).
LGPAS: Ley General de Pesca y Acuacultura Sustentable	Ordenar, fomentar y regular el manejo integral y el aprovechamiento sustentable de la pesca y la acuacultura, considerando los aspectos sociales, tecnológicos, productivos, biológicos y ambientales (Congreso general de los estados Unidos Mexicanos, 2018a).
LGVS: Ley General de Vida Silvestre	El aprovechamiento sustentable de los recursos forestales maderables y no maderables y de las especies cuyo medio de vida total sea el agua, será regulado por las leyes forestales y de pesca, respectivamente, salvo que se trate de especies o poblaciones en riesgo (Congreso general de los estados Unidos Mexicanos, 2018b).
LGDFS: Ley General de Desarrollo Forestal Sustentable	Promover la legalidad en las actividades productivas, mejorar la capacidad de transformación e integración industrial, impulsar la comercialización y fortalecer la organización de redes locales de valor y cadenas productivas del sector forestal (Mexicanos, 2018).

LGPGIR: Ley General Para la Prevención y Gestión Integral de Residuos	Garantizar el derecho de toda persona al medio ambiente sano y propiciar el desarrollo sustentable a través de la prevención de la generación, la valorización y la gestión integral de los residuos peligrosos, de los residuos sólidos urbanos y de manejo especial; prevenir la contaminación de sitios con estos residuos (Mexicanos, 2021).
LFBOGM: Ley Federal de Bioseguridad de Organismos Genéticamente Modificados	Regular las actividades de utilización confinada, liberación experimental, liberación en programa piloto, liberación comercial, comercialización, importación y exportación de organismos genéticamente modificados, con el fin de prevenir, evitar o reducir los posibles riesgos que estas actividades pudieran ocasionar a la salud humana o al medio ambiente y a la diversidad biológica o a la sanidad animal, vegetal y acuícola (Mexicanos, 2020).
LGCC: Ley General De Cambio Climático	Garantizar el derecho a un medio ambiente sano y establecer la concurrencia de facultades de la federación, las entidades federativas y los municipios en la elaboración y aplicación de políticas públicas para la adaptación al cambio climático y la mitigación de emisiones de gases y compuestos de efecto invernadero (Congreso general de los estados Unidos Mexicanos, 2020b).

ANEXO 3. NOMENCLATURA DE LAS FASES Y TAREAS A PARTIR DE CRISP-DM Y TDSP

A continuación, se describe cada una de las fases, tareas y entregables, indicando su identificador, descripción y propósito:

Fases:

BU = Tareas pertenecientes a la fase de Entendimiento del Negocio (*Business Understanding*). Se debe dedicar tiempo a explorar las expectativas de su organización con respecto a la minería de datos, se trata de cómo producir un plan de proyecto utilizando la información que se contiene.

DU = Tareas pertenecientes a la fase de Entendimiento de los Datos (*Data Understanding*). La fase de entendimiento de datos de *CRISP-DM* implica estudiar más de cerca los datos disponibles de minería. Este paso es esencial para evitar problemas inesperados durante la siguiente fase (preparación de datos) que suele ser la fase más larga de un proyecto.

DP = Tareas pertenecientes a la fase de Preparación de los Datos (*Data Preparation*). La preparación de datos es uno de los aspectos más importantes y con frecuencia que más tiempo exigen en la minería de datos.

En cada una de estas fases se describe la tarea a realizar junto con los entregables que de acuerdo a la metodología *TDSP* serán depositados en los repositorios correspondientes a través del uso de plantillas que propone dicha metodología. Dichos entregables están identificados mediante el símbolo ❖.

Las tareas y entregables a considerar en este proyecto para la fase BU se especifican en la Tabla 22.

Tabla 22. Entendimiento del Negocio.

Fases/Tareas	Tipo	Entregables	Carpeta
BU = Entendimiento del Negocio (Business Understanding).	**Fase**	**Documentos o algoritmos**	**Github**
BU-G1 Determinar los objetivos del negocio	Tarea Genérica	• Lista de objetivos del negocio	/Docs/Project
BU-E1 Generar un corpus de leyes ambientales mexicanas	Tarea Especialzada	• Textos de datos destinado a la investigación científica	/Docs/Project

BU-G2 Evaluar la situación	Tarea Genérica	• Lista de recursos computacionales
		  ○ Inventario (Centro de Cómputo y Computadora Personal)
		• Recursos humanos
		  ○ Director del proyecto, codirector del proyecto, investigadoras del IPN con experiencia en PLN
		• Software
		  ○ Python y bibliotecas, Github (Tika, NLTK, PyPDF2 + 2 más)
		• Listado de restricciones
		  ○ 9 leyes
		• Recursos computacionales
		  ○ Tiempo y memoria
			/Docs/Project

BU-G3 Producir el plan del proyecto	Tarea Genérica	• Plan del proyecto
		• Creación del Repositorio de GitHub
		  ○ Con nombre del proyecto
		  ○ Readme (Intro, Colaboración, Estructura)
		  ○ Planificación de carpetas principales
		  ○ Datos (Datos sin procesar, Datos procesados)
		  ○ Docs (Proyecto- Estructura, Reporte de Datos- JSON
		  ○ Código (Preparación de los Datos)
			/Docs/Project

Para el caso de la fase DU = Entendimiento de los Datos (*Data Understanding*), las tareas genéricas a realizar son las siguientes.

Tabla 23. Entendimiento de los Datos.

Fases/Tareas	Tipo	Entregables	Carpeta
DU = Entendimiento de los Datos (Data Understanding).	**Fase**	**Documentos o algoritmos**	**Github**
DU-G1 Recolección de Datos Iniciales	Tarea Genérica	• Reporte de adquisición de datos (describir el proceso de descarga de cada ley)	/Data/Raw
DU-G2 Descripción de los datos	Tarea Genérica	• Estructura de leyes (Logo, nombre, artículos) • Descripción del documento de Ley (Imagen, origen a estructura json)	/Docs/Project
DU-G3 Exploración de los Datos	Tarea Genérica	• Transformación pdf a txt (Python, PyPDF2, pdfplumber) • Reporte de análisis de dichas exploraciones con gráficos generados • Medición de tiempo y memoria	/Docs/Project

Por último, la fase que requiere la fase experimental es DP = Preparación de los Datos (*Data Preparation*), está compuesta de las siguientes tareas genéricas y específicas.

Tabla 24. Preparación de los Datos.

Fases/Tareas	Tipo	Entregables	Carpeta
DP = Preparación de los Datos (Data Preparation).	**Fase**	**Documentos o algoritmos**	**Github**
DP-G1 Descripción del Corpus	Tarea Especializada	• DP-E1. Descripción del formato de ley (formato, hojas) • DP-E2. Transformación pdf a txt con cada herramienta seleccionada.	/Data/Raw
DP-G2 Limpieza de Datos	Tarea Especializada	• DP-E3. Eliminación de encabezados, stopwords, páginas, logos.	/Data/Raw

DP-G3 Formateo	Tarea Especializada	• Las tareas tres y cuatro corresponden a tareas pero no se lograron cumplir en tiempo.	/Data/Raw
DP-G4 Construcción de los datos	Tarea Especializada	• DP-E4. Transformar de TXT a estructura json.	/Data/Raw

ANEXO 4. PLANTILLA PARA EL DISEÑO EXPERIMENTAL

A continuación de describen los elementos que componen la plantilla que servirá como base para la realización de cada uno de los experimentos a efectuar para la fase de preparación de datos. En ésta se incluyen las tareas de formular la pregunta inicial a través de la definición de la hipótesis de cada experimento, la especificación de los elementos que conforman el experimento, así como las herramientas utilizadas para ello.

Los apartados de la plantilla anterior, se explican en la Tabla 6:

Tabla 25. Descripción de la plantilla de diseño experimental.

Rubro	Descripción
Proyecto:	Caracterización de recursos computacionales para la fase de preprocesamiento de mineria de textos
ID_Experimento:	2 letras de la fase genérica,+ 1 dígito del número de fase + _ + 2 letras de la fase especíalizada + _ + consecutivo del número de repetición de 2 dígitos
FechaProg:	Fecha en la que se programa el experimento.
FechaExp:	Fecha en la que se ejecuta el experimento.
Nombre de la tarea:	Descripción corta de la tarea genérica o especializada a probar durante el experimento.
Tareas genéricas:	DP = Preparación de los Datos (Data Preparation). | Tareas especializadas: | TR_01= Transformación de pdf a texto plano
	DP2 Limpieza de datos | | Eliminacion de ruido, Quitar stopwords, paginas, logos, transformar a estructura json
Nombre del pdf:	Nombre de los documentos pdf a procesar.
Bibliotecas:	Nombre de las bibliotecas que se utilizarán en el experimento.
Medición:	Tiempo / Memoria / Ambos.
Wilcoxon:	Determinar si existen diferencias estadísticamente significativas
Tipo de Diseño:	Tipo de diseño el cual es de tipo experimental.
Hipótesis de partida:	Suposición hecha a partir de los datos que sirve para iniciar la investigación y que debe ser probada a partir de la experimentación la cual se tendria que aceptar o rechazar.
Variables independientes:	Variable(s) que no dependerá(n) de otra variable.
Variables dependientes:	Variable(s) cuyos valores dependen de los que tomen las variables independientes.
Variables controladas:	Variable(s) o factore(s) que es (son) fijados o eliminados para identificar claramente la relación entre una variable independiente y una variable dependiente.
Variables no controladas:	Variable(s) que no se manipulan pero pueden influir en el resultado de la investigación.

Resultados esperados:	Metas o resultados que se desean obtener a partir de la experimentación para probar la hipótesis.
Descripción del experimento:	Procedimiento llevado a cabo para apoyar, refutar o validar una hipotesis.
Conclusiones del experimento:	Resumen integral de los resultados, la cual da una visión general del experimento y de si este alcanzó los resultados esperados.

ANEXO 5. TR-01: TRANSFORMACIÓN DEL FORMATO ORIGINAL EN PDF A FORMATO DE TEXTO PLANO

A continuación de describen los elementos que componen la plantilla que servirá como base para la ejecución de esta tarea.

Los scripts utilizados se encuentran en la carpeta **/Data/Raw** dentro de la plantilla ***pypdf2ToText con Medicion.py y pdfplumberToText con Medicion.py***

En concreto, los problemas encontrados con el uso de ambas bibliotecas, es que trabajan con términos en inglés y existe un truncamiento de los mismos durante su conversión. Sin embargo, las bibliotecas PyPDF2 y Pdfplumber para la extracción de pdf a txt demostraron ser adecuadas en los resultados, tiene un mejor preprocesamiento en la transformación de pdf a txt, resulta con mayor cantidad de palabras completas, sin saltos de línea, sin palabras no conocidas, la imagen del texto la transforma a palabras.

La Tabla 27 muestra los resultados del tiempo de ejecución para la tarea de preprocesamiento, la cual incluye, las siguientes fases:

a) transformación de pdf a txt: archivo nombredelaley.txt el cual se encuentra en la carpeta pypdf2 y archivo nombredelaley.txt el cual se encuentra en la carpeta pdfplumber.

Acto seguido, se muestran ejemplos de los diseños experimentales mediante el uso de estas bibliotecas.

Tabla 26. Diseño experimental con la biblioteca PyPDF2.

PLANTILLA DE DISEÑO EXPERIMENTAL			
Nombre Proyecto:	Caracterización de recursos computacionales para la fase de preprocesamiento de mineria de textos		
ID Experimento:	DP_TR_01	**Fecha Programada:**	28/10/2022
		Fecha Experimento:	31/10/2022
Nombre de la tarea:	Transformación de una ley en formato original en pdf a formato de texto plano.		
TIPO DE TAREA			
Fase:	DP = Preparación de los Datos (*Data Preparation*).	**Especializada:**	Transformación pdf a texto plano con cada herramienta seleccionada.
Genérica:	Formateo de datos.		
Nombre del pdf:	LGEEPA: Ley General del Equilibrio Ecológico y la Protección al Ambiente.		
Biblioteca:	PyPDF2.		
Mediciones:	Tiempo y Memoria.		
Factor y sus Niveles:	2 elevado a la 1 x 9= 18		
Prueba estadística:	Wilcoxon.		
	HIPÓTESIS DE PARTIDA	**VARIABLES**	
La variación del tiempo y memoria en la transformación de pdf a txt depende de las bibliotecas utilizadas de *Python*.		**Independientes:**	Tipo de biblioteca en *Python*.
	RESULTADOS ESPERADOS	**Dependientes:**	Tiempo y Memoria.

Obtener un archivo en txt equivalente al original de ley ambiental en pdf.	**Variable(s) Controlada(s):** Características del equipo de cómputo.
	Variable(s) no controlada(s): Daños en el equipo de cómputo.
DESCRIPCIÓN DEL EXPERIMENTO	
Se medirá el tiempo y memoria utilizados en la transformación de 9 textos (leyes ambientales) en formato pdf y con distinto número de páginas, mediante dos distintas bibliotecas de preprocesamiento de textos en *Python* (PyPDF2 / Pdfplumber). El número de repeticiones en cada biblioteca experimentada se calcula de la siguiente manera: 9 documentos de ley multiplicado por las 2 bibliotecas da un total de 18 unidades experimentales El equipo de cómputo donde se ejecutarán las pruebas es el siguiente: Memoria RAM de 3 GB, disco duro de 500 GB, Procesador AMD Procesador 1.60 Ghz x64 y Windows 8.1 Pro. El preprocesamiento se llevará a cabo sin ejecutar otra tarea en la computadora de manera simultánea.	
CONCLUSIONES SOBRE EL EXPERIMENTO	
Las versiones utilizadas en la ejecución de la tarea de preprocesamiento (extracción de texto del archivo pdf) son *Python* 3.10, Pdfplumber 0.1.2 y PyPDFf2 2.10.3. Las bibliotecas PyPDF2 y Pdfplumber para la extracción de pdf a txt mostraron ser adecuadas en los resultados. La biblioteca PyPDF2 introduce saltos de página en cada palabra, cuenta guiones; además las palabras no están completas y no reconoce signos de puntuación. En cambio la biblioteca Pdfplumber cuenta mejor las palabras y no los guiones, saltos de página como palabras y reconoce las letras grandes, los pies de página, diferentes puntos, dos puntos, comas, signos de puntuación, acentos, encabezados e imágenes.	

Tabla 27. Mediciones biblioteca PyPDF2.

PDF A TXT CON PYPDF2						
Ley	**Número de Páginas**	**Tiempo (ms)**	**Memoria del Proceso (bytes)**	**Palabras**	**Memoria RAM**	**Bibliotecas**
Ley de Aguas Nacionales	128	28.6719	78%	58679	2.94	PDF A TXT: from PyPDF2 import PdfFileReader, PdfFileWriter

Ley General del Equilibrio Ecológico y la Protección al Ambiente	112	37.6744	89.6%	61583	2.94	TXT A JSON: import json
Ley de Bioseguridad de Organismos Genéticamente Modificados	73	11.1583	65.9%	26221	2.94	TIEMPO: timeit.default_timer()
Ley de Desarrollo Rural Sustentable	71	14.1313	63.6%	33548	2.94	MEMORIA RAM: import psutil
Ley General del Cambio Climático	72	15.8024	65%	28777	2.94	Memoria del Proceso:Memoria dinámica y utiliza un porcentaje del total de la memoria RAM
Ley General de Pesca y Acuacultura Sustentables	50	19.6925	92.9%	33365	2.94	
Ley General de Vida Silvestre	56	14.2578	87.9%	31825	2.94	
Ley General de Desarrollo Forestal Sustentable	49	17.4321	74.7%	30950	2.94	
Ley General para la Prevención y Gestión Integral de los Residuos	64	43.5912	94.3%	26835	2.94	

Tabla 28. Diseño experimental con la biblioteca Pdfplumber.

PLANTILLA DE DISEÑO EXPERIMENTAL			
Nombre Proyecto:	Caracterización de recursos computacionales para la fase de preprocesamiento de mineria de textos		
ID Experimento:	DP_TR_01	**Fecha Programada:**	28/10/2022
		Fecha Experimento:	31/10/2022
Nombre de la tarea:	Transformación de una ley en formato original en pdf a formato de texto plano.		
TIPO DE TAREA			
Fase:	DP = Preparación de los Datos (*Data Preparation*).	**Especializada:**	Transformación pdf a texto plano con cada herramienta seleccionada.
Genérica:	Formateo de datos.		
Nombre del pdf:	LGEEPA: Ley General del Equilibrio Ecológico y la Protección al Ambiente.		
Biblioteca:	Pdfplumber.		
Mediciones:	Tiempo y Memoria.		
Factor y sus Niveles:	2 elevado a la 1 x 9= 18		
Prueba estadística:	Wilcoxon.		
HIPÓTESIS DE PARTIDA		**VARIABLES**	
La variación del tiempo y memoria en la transformación de pdf a txt depende de las bibliotecas utilizadas de *Python*.		**Independientes:**	Tipo de biblioteca en *Python*.

		RESULTADOS ESPERADOS					**Dependientes:**	Tiempo y Memoria.			
Obtener un archivo en txt equivalente al original de ley ambiental en pdf.							**Variable(s) Controlada(s):**	Características del equipo de cómputo.			
							Variable(s) no controlada(s):	Daños en el equipo de cómputo.			
					DESCRIPCIÓN DEL EXPERIMENTO						
Se medirá el tiempo y memoria utilizados en la transformación de 9 textos (leyes ambientales) en formato pdf y con distinto número de páginas, mediante dos distintas bibliotecas de preprocesamiento de textos en *Python* (PyPDF2 / Pdfplumber). El número de repeticiones en cada biblioteca experimentada se calcula de la siguiente manera: 9 documentos de ley multiplicado por las 2 bibliotecas da un total de 18 unidades experimentales El equipo de cómputo donde se ejecutarán las pruebas es el siguiente: Memoria RAM de 3 GB, disco duro de 500 GB, Procesador AMD Procesador 1.60 Ghz x64 y Windows 8.1 Pro. El preprocesamiento se llevará a cabo sin ejecutar otra tarea en la computadora de manera simultánea.											
					CONCLUSIONES SOBRE EL EXPERIMENTO						
Las versiones utilizadas en la ejecución de la tarea de preprocesamiento (extracción de texto del archivo pdf) son *Python* 3.10, Pdfplumber 0.1.2 y PyPDFf2 2.10.3. Las bibliotecas PyPDF2 y Pdfplumber para la extracción de pdf a txt mostraron ser adecuadas en los resultados. La biblioteca PyPDF2 introduce saltos de página en cada palabra, cuenta guiones; además las palabras no están completas y no reconoce signos de puntuación. En cambio la biblioteca Pdfplumber cuenta mejor las palabras y no los guiones, saltos de página como palabras y reconoce las letras grandes, los pies de página, diferentes puntos, dos puntos, comas, signos de puntuación, acentos, encabezados e imágenes.											

Tabla 29. Mediciones biblioteca Pdfplumber.

PDF A TXT CON PDFPUMBER						
Ley	**Número de Páginas**	**Tiempo (ms)**	**Memoria del Proceso (bytes)**	**Palabras**	**Memoria RAM**	**Bibliotecas**
Ley de Aguas Nacionales	128	113.4356	89%	55477	2.94	PDF A TXT: import pdfplumber

Ley General del Equilibrio Ecológico y la Protección al Ambiente	112	147.1395	94.5%	55737	2.94	TXT A JSON: import json
Ley de Bioseguridad de Organismos Genéticamente Modificados	73	44.9475	76.4%	25061	2.94	TIEMPO: timeit.default_timer()
Ley de Desarrollo Rural Sustentable	71	57.6075	78.8%	31460	2.94	MEMORIA RAM: import psutil
Ley General del Cambio Climático	72	53.6704	79%	26844	2.94	Memoria del Proceso:Memoria dinámica y utiliza un porcentaje del total de la memoria RAM
Ley General de Pesca y Acuacultura Sustentables	50	61.8336	94.7%	31396	2.94	
Ley General de Vida Silvestre	56	123.8178	93.2%	29454	2.94	
Ley General de Desarrollo Forestal Sustentable	49	58.2579	91.4%	29822	2.94	
Ley General para la Prevención y Gestión Integral de los Residuos	64	47.1765	84%	24995	2.94	

ANEXO 6. TR-02: CONVERSIÓN DEL FORMATO DE TEXTO PLANO (TXT) A FORMATO JSON

En este apartado, se muestran las actividades realizadas para llevar a cabo la segunda tarea en la cual el texto resultante de pdf a txt se convierte a json.

Las características de dichos documentos, hicieron que los recursos computacionales tuvieran variaciones. Un valor fijo utilizado fue el número de hojas en los documentos, se pensó que resultaría una hipótesis nula la cual es: que los tiempos y memorias son iguales en todos los documentos, pero resulto la hipótesis alternativa la cual es: que hay diferencia en los tiempos (milisegundos) y en memoria (bytes) de cada uno de los documentos.

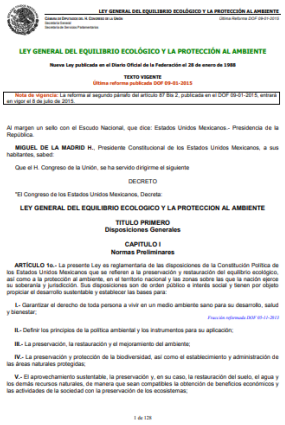
En esta tarea general las dos bibliotecas hicieron el mismo proceso de transformación de pdf a txt, en las cuales se quiere ver cuál es la mejor biblioteca para seguir con la siguiente tarea de conversión de txt a json, los cuales los archivos txt generados de la primer tarea de transforman en json pero se tienen que ver la estructura, patrones de comportamiento entre los textos para asi hacer una limpieza de estos mismos con scripts para finalmente convertirlos a formato json.

Patrones de los pdf

A continuación se observa el nombre de la ley, la figura del pdf de dicha ley, asi como los distintos patrones que se tienen en cada uno de los 9 documentos de leyes ambientales.

Tabla 30. Patrones que contienen los pdf.

LGEEPA: Ley General del Equilibrio Ecológico y la Protección al Ambiente	Encabezado El encabezado comienza con LEY Sigue dos guiones largos



LEY GENERAL DEL EQUILIBRIO ECOLÓGICO Y LA PROTECCIÓN AL AMBIENTE

Última Reforma DOF 09-01-2015

Cámara de Diputados del H. Congreso de la Unión
Secretaría General
Secretaría de Servicios Parlamentarios

LEY GENERAL DEL EQUILIBRIO ECOLÓGICO Y LA PROTECCIÓN AL AMBIENTE

Nueva Ley publicada en el Diario Oficial de la Federación el 28 de enero de 1988

TEXTO VIGENTE

Última reforma publicada DOF 09-01-2015

Nota de vigencia: La reforma al segundo párrafo del artículo 87 Bis 2, publicada en el DOF 09-01-2015, entrará en vigor el 8 de julio de 2015.

Al margen un sello con el Escudo Nacional, que dice: Estados Unidos Mexicanos.- Presidencia de la República.

MIGUEL DE LA MADRID H., Presidente Constitucional de los Estados Unidos Mexicanos, a sus habitantes, sabed:

Que el H. Congreso de la Unión, se ha servido dirigirme el siguiente

DECRETO

"El Congreso de los Estados Unidos Mexicanos, Decreta:

LEY GENERAL DEL EQUILIBRIO ECOLOGICO Y LA PROTECCION AL AMBIENTE

TITULO PRIMERO
Disposiciones Generales

CAPITULO I
Normas Preliminares

ARTÍCULO 1o.- La presente Ley es reglamentaria de las disposiciones de la Constitución Política de los Estados Unidos Mexicanos que se refieren a la preservación y restauración del equilibrio ecológico, así como a la protección al ambiente, en el territorio nacional y las zonas sobre las que la nación ejerce su soberanía y jurisdicción. Sus disposiciones son de orden público e interés social y tienen por objeto propiciar el desarrollo sustentable y establecer las bases para:

I.- Garantizar el derecho de toda persona a vivir en un medio ambiente sano para su desarrollo, salud y bienestar;

Fracción reformada DOF 07-11-2013

II.- Definir los principios de la política ambiental y los instrumentos para su aplicación;

III.- La preservación, la restauración y el mejoramiento del ambiente;

IV.- La preservación y protección de la biodiversidad, así como el establecimiento y administración de las áreas naturales protegidas;

V.- El aprovechamiento sustentable, la preservación y, en su caso, la restauración del suelo, el agua y los demás recursos naturales, de manera que sean compatibles la obtención de beneficios económicos y las actividades de la sociedad con la preservación de los ecosistemas;

1 de 128

Parte izquierda el logo

Después del guión sigue:

Cámara de Diputados del H. Congreso de la Unión con espacio y sigue Última Reforma DOF 09-01-2015

Siguiente línea Secretaría General

Siguiente Secretaria de Servicios Parlamentarios

Título

Comienza con LEY

Siguiente línea: Nueva Ley publicada en el Diario Oficial de la Federación el 28 de enero de 1988

Siguiente Línea: TEXTO VIGENTE

Siguiente Línea: Última reforma publicada DOF 09-01-2015

Siguiente Línea sigue un cuadro que menciona: Nota de vigencia: La reforma al segundo párrafo del artículo 87 Bis 2, publicada en el DOF 09-01-2015, entrará en vigor el 8 de julio de 2015.

Contenido: Al margen un sello con el Escudo Nacional, que dice: Estados Unidos Mexicanos.- Presidencia de la República.

Siguiente línea: MIGUEL DE LA MADRID H., Presidente Constitucional de los Estados Unidos Mexicanos, a sus habitantes, sabed:

Siguiente línea: Que el H. Congreso de la Unión, se ha servido dirigirme el siguiente

Siguiente línea: DECRETO

Siguiente línea: "El Congreso de los Estados Unidos Mexicanos, Decreta:

Título de la LEY

Título Primero

	Siguiente Línea: Disposiciones Generales Capitulo Comienza con: CAPÍTULO I Siguiente Línea: Título de ese capítulo Artículo ARTÍCULO 16.- Las Título Segundo Capítulo Comienza con: CAPÍTULO Siguiente Línea: Título de ese capítulo Sección Comienza: SECCIÓN I Siguiente Línea: Título de esa sección Artículo Comienza con: ARTÍCULO 1o.- Siguiente Línea: I.- Garantizar el derecho de toda persona a vivir en un medio ambiente sano para su desarrollo, salud y bienestar; Siguiente Línea: Fracción reformada DOF 05-11-2013 la cual se repite muchas veces en el documento. Finaliza con el número de página: 1 de 128 Transitorio Se repiten: Artículo reformado DOF 13-12-1996 Fracción reformada DOF 13-12-1996

	Fracción reformada DOF 13-12-1996, 28-01-2011 Fracción reformada DOF 28-01-2011 Fracción adicionada DOF 28-01-2011 Fracción adicionada DOF 28-01-2011 Fracción adicionada DOF 28-01-2011 Fracción recorrida DOF 28-01-2011 Fracción reformada DOF 01-04-2010. Recorrida DOF 28-01-2011 Fracción recorrida DOF 28-01-2011. Reformada DOF 16-01-2014 Fracción reformada DOF 07-01-2000. Recorrida DOF 28-01-2011. Reformada DOF 09-04-2012 Fracción adicionada DOF 04-06-2012 Fracción reformada DOF 07-01-2000. Recorrida DOF 28-01-2011, 04-06-2012 Fracción adicionada DOF 07-01-2000. Recorrida DOF 28-01-2011, 04-06-2012 Fracción adicionada DOF 23-02-2005. Recorrida DOF 28-01-2011, 04-06-2012 Artículo reformado DOF 13-12-1996 Denominación del Capítulo reformada DOF 13-12-1996 Párrafo adicionado DOF 25-02-2003 Artículo reformado DOF 13-12-1996 Fracción reformada DOF 25-02-2003 Fracción reformada DOF 28-01-2011 Fracción adicionada DOF 28-01-2011 Fracción recorrida DOF 28-01-2011 Artículo reformado DOF 13-12-1996

	Párrafo reformado DOF 23-05-2006 Fracción reformada DOF 28-01-2011 Fracción adicionada DOF 28-01-2011 Fracción recorrida DOF 28-01-2011 Artículo reformado DOF 13-12-1996 Artículo reformado DOF 13-12-1996, 31-12-2001 Artículo adicionado DOF 13-12-1996 Denominación del Capítulo reformada DOF 13-12-1996 (reubicado) Fracción reformada DOF 24-04-2012 Artículo adicionado DOF 15-05-2013 Denominación de la Sección reformada DOF 13-12-1996 Párrafo reformado DOF 13-12-1996 Fracción reformada DOF 13-12-1996 Fracción reformada DOF 12-02-2007 Fracción reformada DOF 13-12-1996, 12-02-2007 Fracción adicionada DOF 12-02-2007 Artículo adicionado DOF 13-12-1996 Párrafo adicionado DOF 12-02-2007 Artículo adicionado DOF 13-12-1996 Denominación de la Sección reformada DOF 13-12-1996 Párrafo reformado DOF 29-05-2012 Fracción recorrida DOF 05-07-2007 Párrafo reformado DOF 23-02-2005

	Fracción derogada DOF 25-02-2003 Artículo reformado DOF 13-12-1996, 20-05-2013 Artículo adicionado DOF 13-12-1996. Recorrido (antes Artículo 37 BIS) DOF 24-05-2013 Inciso reformado DOF 24-05-2013 Denominación del Capítulo reformada DOF 13-12-1996 (se recorre, antes Capítulo VI) Denominación del Capítulo reformada DOF 13-12-1996 (se recorre, antes Capítulo VII)
LAN: Ley de Aguas Nacionales  LEY DE AGUAS NACIONALES Nueva Ley publicada en el Diario Oficial de la Federación el 1º de diciembre de 1992 TEXTO VIGENTE Última reforma publicada DOF 06-01-2020 Al margen un sello con el Escudo Nacional, que dice: Estados Unidos Mexicanos.- Presidencia de la República. CARLOS SALINAS DE GORTARI, Presidente Constitucional de los Estados Unidos Mexicanos, a sus habitantes sabed: Que el H. Congreso de la Unión, se ha servido dirigirme el siguiente D E C R E T O "EL CONGRESO DE LOS ESTADOS UNIDOS MEXICANOS, D E C R E T A: LEY DE AGUAS NACIONALES TÍTULO PRIMERO Disposiciones Preliminares Capítulo Único ARTÍCULO 1. La presente Ley es reglamentaria del Artículo 27 de la Constitución Política de los Estados Unidos Mexicanos en materia de aguas nacionales; es de observancia general en todo el territorio nacional, sus disposiciones son de orden público e interés social y tiene por objeto regular la explotación, uso o aprovechamiento de dichas aguas, su distribución y control, así como la preservación de su cantidad y calidad para lograr su desarrollo integral sustentable. ARTÍCULO 2. Las disposiciones de esta Ley son aplicables a todas las aguas nacionales, sean superficiales o del subsuelo. Estas disposiciones también son aplicables a los bienes nacionales que la presente Ley señala. Las disposiciones de esta Ley son aplicables a las aguas de zonas marinas mexicanas en tanto a la conservación y control de su calidad, sin menoscabo de la jurisdicción o concesión que las pudiere regir. ARTÍCULO 3. Para los efectos de esta Ley se entenderá por: I. "Aguas Nacionales": Son aquellas referidas en el Párrafo Quinto del Artículo 27 de la Constitución Política de los Estados Unidos Mexicanos; II. "Acuífero": Cualquier formación geológica o conjunto de formaciones geológicas hidráulicamente conectados entre sí, por las que circulan o se almacenan aguas del subsuelo que pueden ser extraídas para su explotación, uso o aprovechamiento y cuyos límites laterales y verticales se definen convencionalmente para fines de evaluación, manejo y administración de las aguas nacionales del subsuelo; 1 de 112	Encabezado El encabezado comienza con LEY Sigue dos guiones largos Parte izquierda el logo Después del guión sigue: Cámara de Diputados del H. Congreso de la Unión con espacio y sigue Última Reforma DOF 06-01-2020 Siguiente línea Secretaría General Siguiente Secretaria de Servicios Parlamentarios Título Comienza con LEY Siguiente línea: Nueva Ley publicada en el Diario Oficial de la Federación el 1º de diciembre de 1992 Siguiente Línea: TEXTO VIGENTE Siguiente Línea: Última reforma publicada DOF 06-01-2020 Siguiente Línea: NO CONTIENE CUADRO

	Contenido: Al margen un sello con el Escudo Nacional, que dice: Estados Unidos Mexicanos.- Presidencia de la República. Siguiente línea: CARLOS SALINAS DE GORTARI, Presidente Constitucional de los Estados Unidos Mexicanos, a sus habitantes sabed: Siguiente línea: Que el H. Congreso de la Unión, se ha servido dirigirme el siguiente Siguiente línea: DECRETO Siguiente línea: "EL CONGRESO DE LOS ESTADOS UNIDOS MEXICANOS, D E C R E T A : Título de la LEY Título Primero Capítulo Comienza con: CAPÍTULO I Siguiente Línea: Título de ese capitulo Artículo ARTÍCULO 16.- Las Título Segundo Capítulo Comienza con: CAPÍTULO Siguiente Línea: Título de ese capítulo Sección Comienza: SECCIÓN I Siguiente Línea: Título de esa sección

	Artículo Comienza con: ARTÍCULO 1o.- Finaliza con el número de página: 1 de 112 Transitorio Se repite: Artículo reformado DOF 29-04-2004 Fracción adicionada DOF 24-03-2016 Inciso reformado DOF 08-06-2012 Fracción reformada DOF 24-03-2016 Fracción adicionada DOF 11-08-2014 Fracción adicionada DOF 20-06-2011 Artículo adicionado DOF 29-04-2004 Capítulo adicionado DOF 29-04-2004 Artículo reformado DOF 29-04-2004 Capítulo reubicado DOF 29-04-2004 Fracción reformada DOF 08-06-2012 Artículo adicionado DOF 29-04-2004 Fracción reformada DOF 08-06-2012 Fracción reformada DOF 07-06-2013 Sección adicionada DOF 29-04-2004 Capítulo adicionado DOF 29-04-2004. Denominación reformada DOF 08-06-2012 Fe de erratas al artículo DOF 15-02-1993. Reformado DOF 29-04-2004

	Párrafo reformado DOF 08-06-2012 Denominación del Título reformada DOF 29-04-2004 Capítulo recorrido (antes Capítulo I) DOF 08-06-2012 Artículo derogado DOF 29-04-2004

Los patrones anteriores en cada pdf es de importancia ya que muestra las posibles incoherencias, errores, repeticiones, secciones, saltos de páginas, pies de páginas, encabezados y diversas redundancias que se necesitan eliminar para tener un documento limpio.

Limpieza de los documentos

Para la limpieza de estos documentos en general se tienen las siguientes caracteristicas:

- Todos los encabezados comienzan con LEY
- Todas las leyes contienen dos guiones largos
- Todos tienen en la parte izquierda el logo
- Después del guión sigue:
- Cámara de Diputados del H. Congreso de la Unión con espacio y sigue un texto el cual no es igual en todas ejemplo: Última Reforma DOF 18-01-2021
- Todas las leyes contienen Secretaría General
- Todas las leyes contienen Secretaria de Servicios Parlamentarios
- Después de encabezado sigue el TÍTULO DE LA LEY
- Siguiente línea un ejemplo: Nueva Ley publicada en el Diario Oficial de la Federación el 8 de octubre de 2003
- Todas las leyes contienen: TEXTO VIGENTE
- Solo una ley no contiene: Última reforma publicada DOF 18-01-2021

- Solo dos leyes contienen un cuadro
- Todas las leyes comienzan con el contenido: Al margen un sello con el Escudo Nacional, que dice: Estados Unidos Mexicanos.- Presidencia de la República.
- Todas las leyes tienen el texto pero cambia el nombre del presidente en el año de la ley: VICENTE FOX QUESADA, Presidente de los Estados Unidos Mexicanos, a sus habitantes sabed:
- Todas las leyes dicen: Que el Honorable Congreso de la Unión, se ha servido dirigirme el siguiente
- Todas las leyes contienen: DECRETO
- Todas las leyes contienen: "EL CONGRESO GENERAL DE LOS ESTADOS UNIDOS MEXICANOS, D E C R E T A:

De ahí lo siguiente es el **Título de la LEY y** algunos títulos tienen más palabras refiriéndose más a lo que trata la ley

Sigue:

Título Primero

Capítulo

Comienza con: CAPÍTULO I

Siguiente Línea: Título de ese capítulo

Artículo

ARTÍCULO 16.- Las

Título Segundo

Capítulo

Comienza con: CAPÍTULO

Siguiente Línea: Título de ese capítulo

Los capítulos suelen estar dentro del título, pero a veces no, identificar los encabezados y pies de página de cada documento.

Dentro del capítulo podemos encontrar artículos que son la división fundamental de las leyes.

Sección

Comienza: SECCIÓN I

Siguiente Línea: Título de esa sección

Artículo

Comienza con: ARTÍCULO 1o.-

Finaliza con el número de página: 1 de 56

Transitorio

Las leyes tienen divisiones donde el título y los artículos transitorios se consideraron de segundo nivel, cada título tiene un número y un nombre

Eliminar letras azules que aparecen:

Párrafo reformado DOF 13-07-2018

Fracción reformada DOF 13-07-2018

Fracción adicionada DOF 13-07-2018

Fracción recorrida DOF 13-07-2018. Derogada DOF 06-11-2020

Artículo reformado DOF 19-01-2018

Inciso reformado DOF 13-05-2015

Inciso reformado DOF 06-11-2020

Párrafo reformado DOF 29-12-2014

Párrafo adicionado DOF 13-07-2018

Fracción reformada DOF 01-06-2016

Fracción reformada DOF 13-05-2015

Estructura de las leyes ambientales

A continuación se muestra el cuerpo que compone las leyes ambientales.

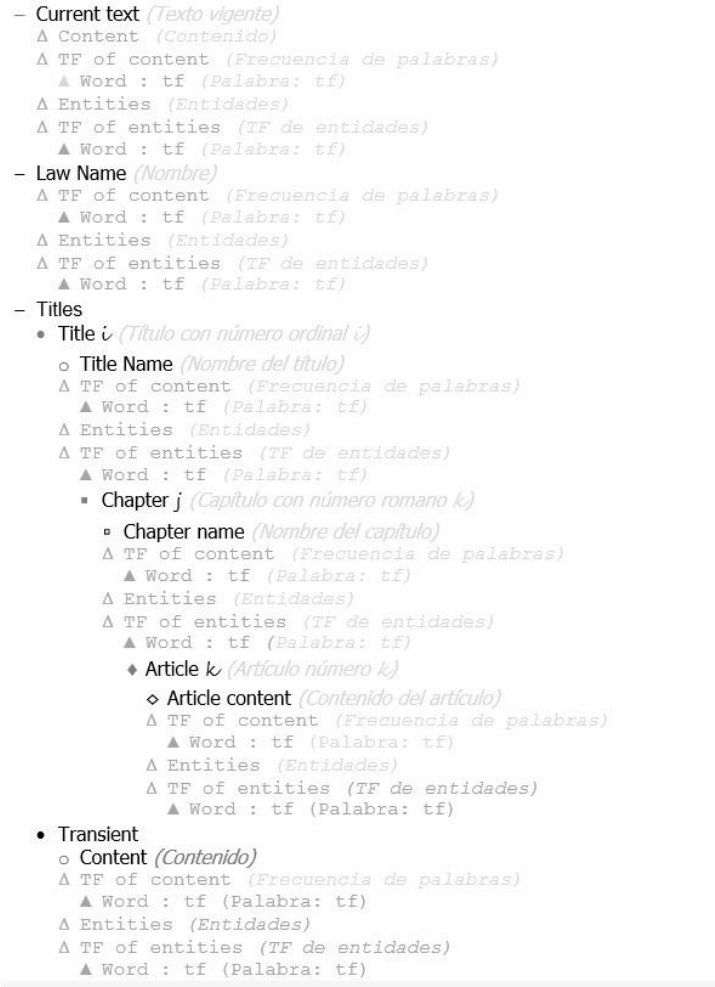


Figura 20. Estructura de cada ley con frecuencias de términos del texto original (Martinez-Seis et al., 2022).

Deconstrucción de las leyes ambientales

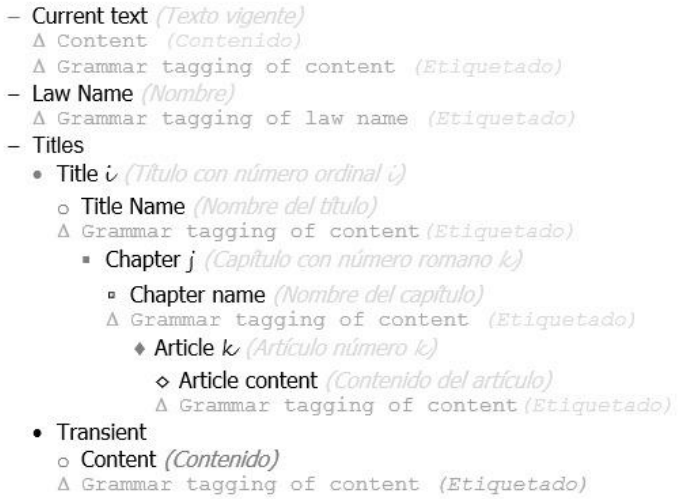


Figura 21. Deconstrucción de la jerarquía interna de las divisiones de las leyes para la representación de documentos (Martinez-Seis et al., 2022).

Cada uno de los elementos de esta deconstrucción jerárquica de las leyes se describe con mayor detalle a continuación:

- **Texto Actual *(Texto Vigente)***. Descripción general incluida en la ley original como: fecha de publicación de la última reforma, nombre del presidente, elementos secundarios como encabezados y descripción en texto de sellos en portadas o márgenes.
- **Nombre de la ley *(Nombre)***. Cada ley tiene un nombre. En muy pocos casos, el número y nombre del libro se encontrará también en esta sección; como sólo las leyes muy extensas se dividen en libros que tratan de recoger una sola rama del derecho, en este caso habrá un documento por cada libro de la ley.
- **Títulos *(Títulos)*.** Algunas leyes se dividen en títulos con partes claramente diferenciadas. Esta división se presenta en leyes extensivas y generales.

- **Título** *i* **(Título** *i***)**. Esta sección tiene la palabra *Título* y su número ordinal.
- **Título Nombre *(Nombre del título)***. Esta etiqueta tiene el nombre asignado al título.
- **Capítulo** *l* ***(Capítulo)***. Es la división más común en las leyes. En general, los capítulos pueden o no estar dentro de los títulos. Están numerados con números romanos.
- **Nombre del capítulo *(Nombre del capítulo)***. Cada capítulo tiene un nombre corto.
  - **Artículo** *k* **(Artículo** *k***)**. El artículo es la división elemental y fundamental de las leyes; incluye una disposición legal condensada en una sola o varias oraciones, a veces dividida en varios párrafos. Cada artículo debe regular un solo tema o precepto. En esta deconstrucción, la etiqueta del artículo incluye todos los párrafos *(párrafos)* , secciones *(apartados)* , fracciones *(fracciones)* y subsecciones *(incisos)* de ese artículo.
- **Transitorios *(Transitorios)***. Disposición destinada a regular situaciones transitorias que existen con anterioridad a la vigencia de una ley.

Algunos desafíos para la transformación automática de texto sin formato a la estructura generalizada en json fueron:

- Falta de homogeneización de la estructura del documento no solo por el tiempo sino también por el mismo gobierno. Algunas leyes carecen de secciones o las incluyen en un orden jerárquico diferente.
- Faltas de ortografía como acentos ortográficos y abreviaturas.
- La numeración de un mismo elemento puede ser números cardinales u ordinales.

- Los artículos especiales como los artículos derogados y los artículos transitorios, que son artículos que se utilizan cuando surgen o cambian nuevas normas; por lo tanto, son efectivos durante la transición.

Para realizar esta deconstrucción se procede a instalar Apache Tika.

Instalando Apache Tika

Para la limpieza de estos documentos, se utiliza el kit de herramientas Apache Tika detecta y extrae metadatos y texto de más de mil tipos de archivos diferentes (como PPT, XLS y PDF). Todos estos tipos de archivos se pueden analizar a través de una sola interfaz, lo que hace que Tika sea útil para la indexación de motores de búsqueda, el análisis de contenido, la traducción y mucho más (https://tika.apache.org/).

Se tuvo que instalar *Java* para utilizar esta herramienta, asi como poder manejarlo para el lenguaje de programación *Python* con expresiones regulares las cuales son un método por medio del cual se pueden realizar búsquedas dentro de cadenas de caracteres requerida de un patrón definido.

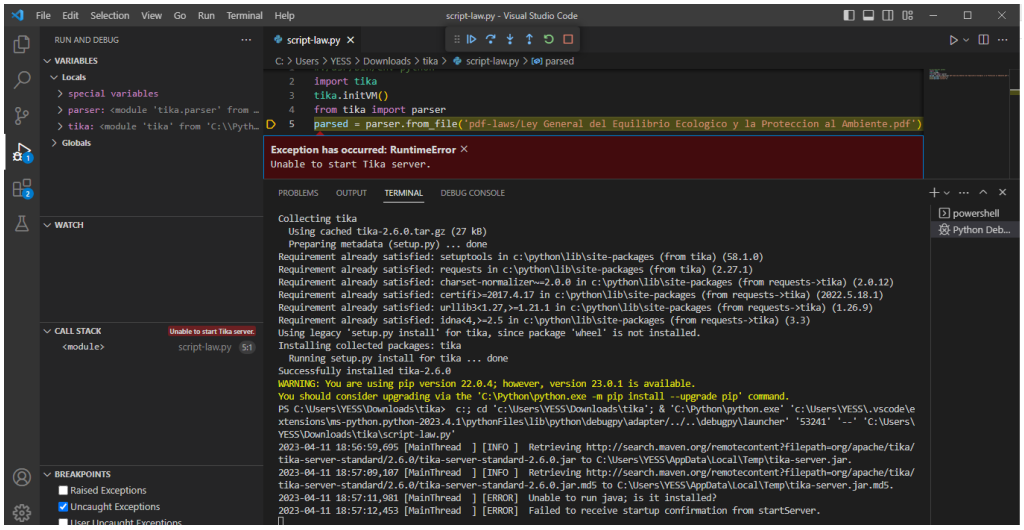


Figura 22. Instalación de Apache Tika (Parte 1).

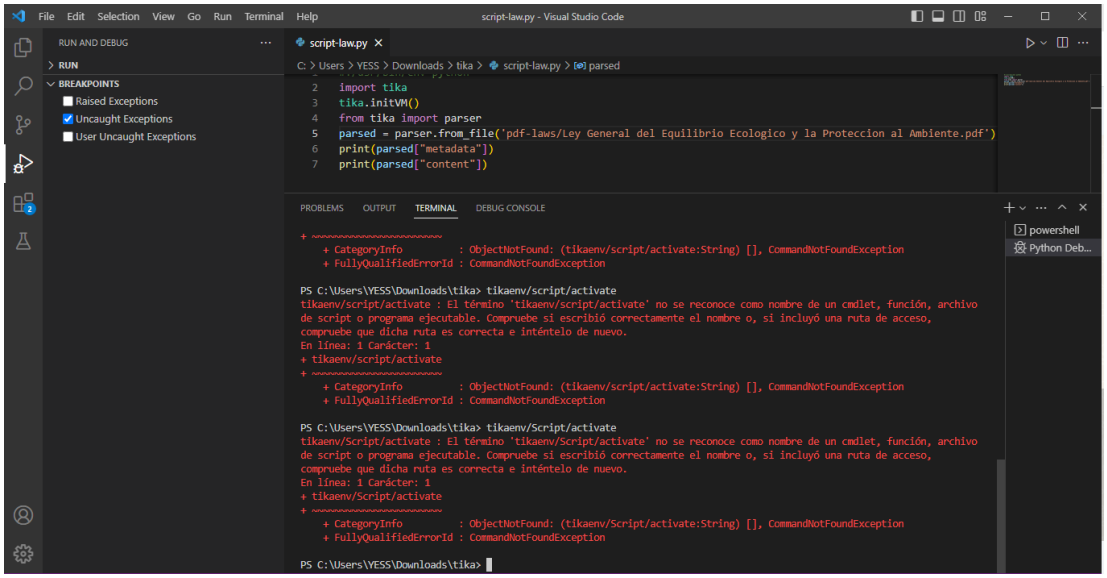


Figura 23. Instalación de Apache Tika (Parte 2).

Spyder (*Python*)

Para la visualización de estos *script* se utilizó el entorno gráfico de *Spyder* para analistas de datos.

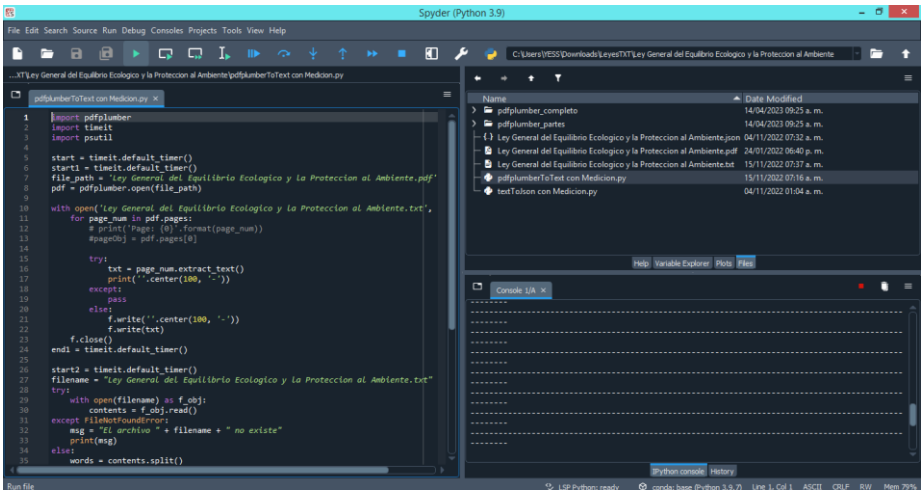


Figura 24. Visualización del script (Parte 1).

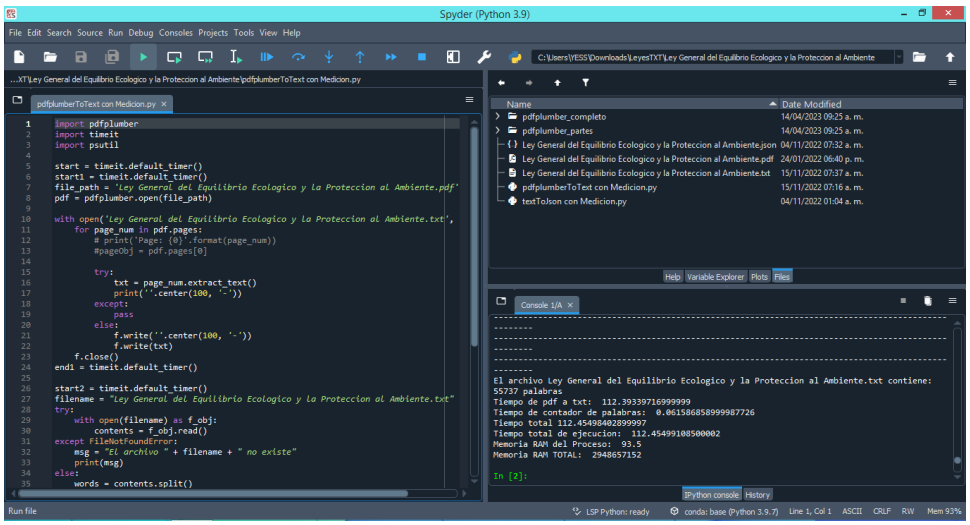


Figura 25. Visualización del script (Parte 2).

Txt resultante

El texto plano resultante son el número de encabezados que contiene cada ley, en este caso cada página que contiene la ley es el número de encabezado que contiene.

```
PS C:\Users\YESS\Downloads\LeyesTXT> python .\TextoVigente.py
ARCHIVO:  Ley de Aguas Nacionales.txt
Numero de encabezados encontrados:  112
ARCHIVO:  Ley de Bioseguridad de Organismos Geneticamente Modificados.txt
Numero de encabezados encontrados:  49
ARCHIVO:  Ley de Desarrollo Rural Sustentable.txt
Numero de encabezados encontrados:  73
ARCHIVO:  Ley General de Cambio Climatico.txt
Numero de encabezados encontrados:  64
ARCHIVO:  Ley General de Desarrollo Forestal Sustentable.txt
Numero de encabezados encontrados:  50
ARCHIVO:  Ley General de Pesca y Acuacultura Sustentables.txt
Numero de encabezados encontrados:  71
ARCHIVO:  Ley General de Vida Silvestre.txt
Numero de encabezados encontrados:  72
ARCHIVO:  Ley General del Equilibrio Ecologico y la Proteccion al Ambiente.txt
Numero de encabezados encontrados:  128
ARCHIVO:  Ley General para la Prevencion y Gestion Integral de los Residuos.txt
Numero de encabezados encontrados:  56
Numero de archivos es:  9
```

Figura 26.Texto resultante (Parte 1).

```
El encabezado es:


--------------------------------------------------------------------------------
LEY DE AGUAS NACIONALES

CÁMARA DE DIPUTADOS DEL H. CONGRESO DE LA UNIÓN  Última Reforma DOF 06-01-2020
Secretaría General
Secretaría de Servicios Parlamentarios


Numero de encabezados encontrados: 112


El encabezado es:


--------------------------------------------------------------------------------
LEY DE BIOSEGURIDAD DE ORGANISMOS GENÉTICAMENTE MODIFICADOS

CÁMARA DE DIPUTADOS DEL H. CONGRESO DE LA UNIÓN  Última Reforma DOF 06-11-2020
Secretaría General
Secretaría de Servicios Parlamentarios


Numero de encabezados encontrados: 49
```

Figura 27. Texto resultante (Parte 2).

LEY GENERAL DEL EQUILIBRIO ECOLÓGICO Y LA PROTECCIÓN AL AMBIENTE

CÁMARA DE DIPUTADOS DEL H. CONGRESO DE LA UNIÓN Última Reforma DOF 09-01-2015
Secretaría General
Secretaría de Servicios Parlamentarios

LEY GENERAL DEL EQUILIBRIO ECOLÓGICO Y LA PROTECCIÓN AL AMBIENTE

Nueva Ley publicada en el Diario Oficial de la Federación el 28 de enero de 1988

TEXTO VIGENTE
Última reforma publicada DOF 09-01-2015

Nota de vigencia: La reforma al segundo párrafo del artículo 87 Bis 2, publicada en el DOF 09-01-2015, entrará en vigor el 8 de julio de 2015.

Al margen un sello con el Escudo Nacional, que dice: Estados Unidos Mexicanos.- Presidencia de la República.

MIGUEL DE LA MADRID H., Presidente Constitucional de los Estados Unidos Mexicanos, a sus habitantes, sabed:

Que el H. Congreso de la Unión, se ha servido dirigirme el siguiente

DECRETO

Figura 28. Texto resultante (Parte 3).

Análisis txt resultante

- Se tienen en todas las leyes guiones ------
- Todos los encabezados contienen la palabra LEY
- Ya no aparece el LOGO
- Todas las leyes contienen Cámara de Diputados del H. Congreso de la Unión
- 8 leyes excepto 1 contienen diferentes palabras similares a ejemplo: Última Reforma DOF 18-01-2021 Todas las leyes contienen Secretaría General
- La Ley General de Desarrollo Forestal Sustentable contiene el texto diferente a ejemplo: Nueva Ley DOF 05-06-2018 solo aparece en esta
- Todas las leyes contienen Secretaria de Servicios Parlamentarios
- Todas las leyes después de encabezado sigue el TÍTULO DE LA LEY
- Todas las leyes contienen diferentes palabras ejemplo: Nueva Ley publicada en el Diario Oficial de la Federación el 8 de octubre de 2003
- Todas las leyes contienen: TEXTO VIGENTE
- En 2 leyes que son Ley General del Equilibrio Ecológico y la Protección al Ambiente y Ley General de Desarrollo Forestal Sustentable, contienen un cuadro en el cual el pdf se elimina pero contiene una Nota de Vigencia

- Todas las leyes comienzan con el contenido: Al margen un sello con el Escudo Nacional, que dice: Estados Unidos Mexicanos.- Presidencia de la República.
- Todas las leyes tienen el texto pero cambia el nombre del presidente en el año de la ley: VICENTE FOX QUESADA, Presidente de los Estados Unidos Mexicanos, a sus habitantes sabed:
- Todas las leyes dicen: Que el Honorable Congreso de la Unión, se ha servido dirigirme el siguiente
- Todas las leyes contienen: DECRETO
- Todas las leyes contienen: "EL CONGRESO GENERAL DE LOS ESTADOS UNIDOS MEXICANOS, D E C R E T A:

De ahí lo siguiente es el **Título de la LEY y** algunos títulos tienen más palabras refiriéndose más a lo que trata la ley

Sigue:

Título Primero

Capitulo

Comienza con: CAPITULO I

Siguiente Línea: Título de ese capitulo

Artículo

ARTÍCULO 16.- Las

Título Segundo

Capitulo

Comienza con: CAPITULO

Siguiente Línea: Título de ese capitulo

Los capítulos suelen estar dentro del título, pero a veces no, identificar los encabezados y pies de página de cada documento.

Dentro del capítulo podemos encontrar artículos que son la división fundamental de las leyes.

Sección

Comienza: SECCIÓN I

Siguiente Línea: Título de esa sección

Artículo

Comienza con: ARTÍCULO 1o.-

Finaliza con el número de página: 1 de 56

Todas las leyes tienen un artículo transitorio que se consideran de segundo nivel, cada título tiene un número y un nombre.

Todas las leyes contienen letras ejemplo:

Párrafo reformado DOF 13-07-2018

Fracción reformada DOF 13-07-2018

Fracción adicionada DOF 13-07-2018

Fracción recorrida DOF 13-07-2018. Derogada DOF 06-11-2020

Artículo reformado DOF 19-01-2018

ANEXO 7. IMPLEMENTACIÓN EN LA NUBE

A continuación, se muestra la creación e implementación de instancia en la nube.

Creación de la instancia

Los resultados de la creación de la nueva instancia en *AWS* se muestran a través de las siguientes imágenes:

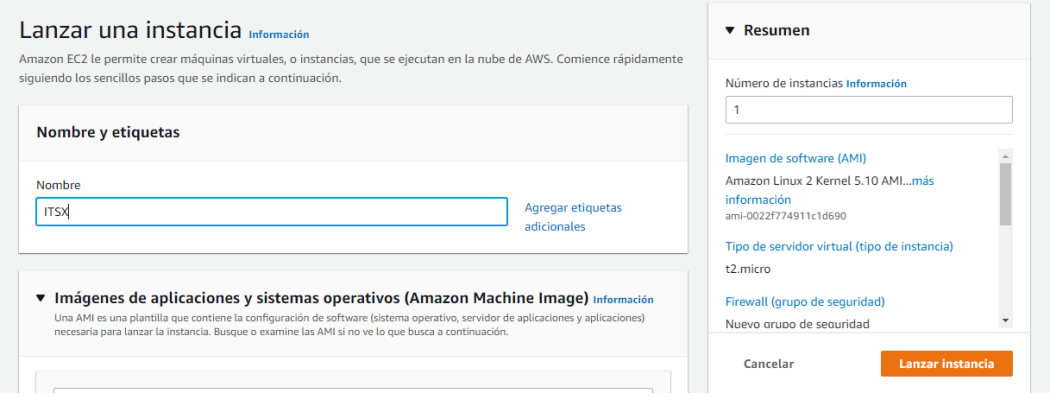


Figura 29. Creación de la instancia (Parte 1).

Se selecciona la instancia *Amazon Linux*.

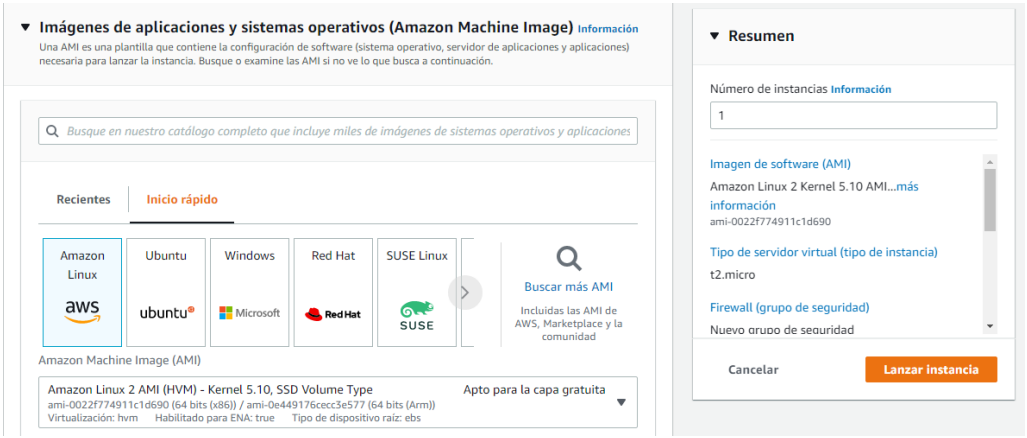


Figura 30. Creación de la instancia (Parte 2).

Se muestra la descripción de la arquitectura la cual contiene un *kernel* 5.10.

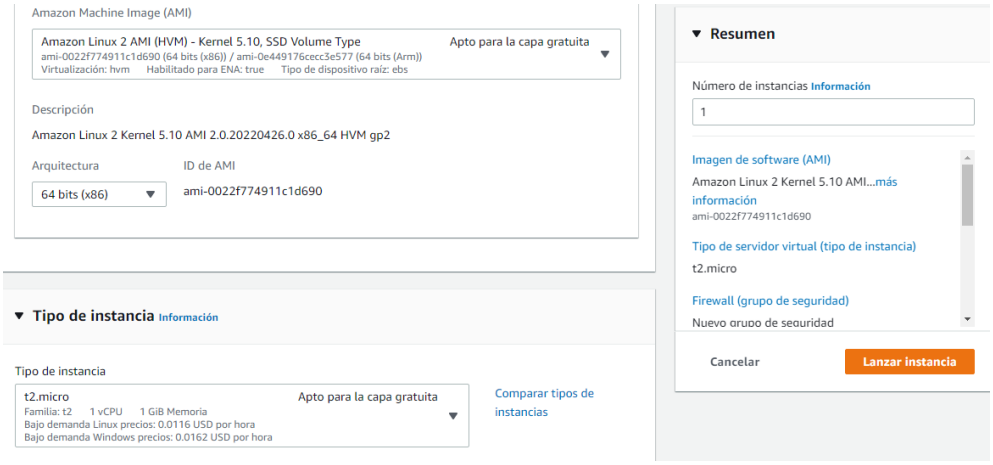


Figura 31. Creación de la instancia (Parte 3).

Se muestra el tipo de instancia la cual es t2.micro.

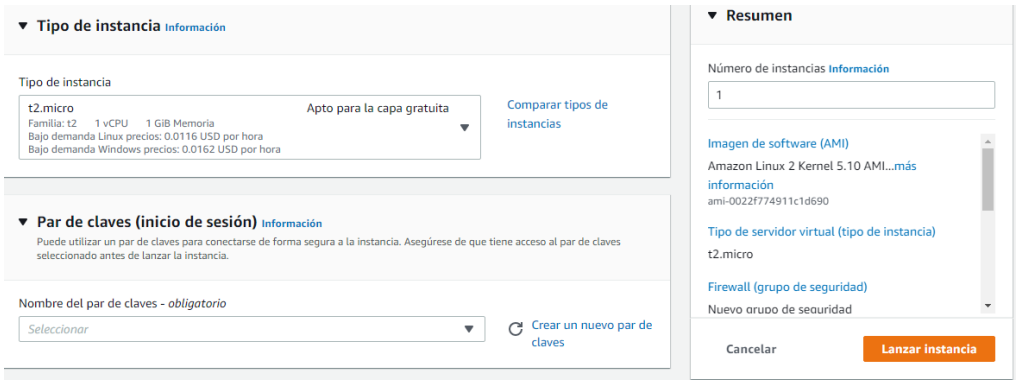


Figura 32. Creación de la instancia (Parte 4).

Se muestra al inicio de sesión un par de claves:

- AWS utiliza criptografía de clave pública para proteger la información de inicio de sesión de la instancia.
- Una instancia de Linux no tiene contraseñas; en su lugar, se utiliza un par de claves para iniciar sesión en su instancia de manera segura.

- Especifica el nombre del par de claves cuando lanza la instancia, luego proporciona la clave privada cuando inicia sesión con SSH.

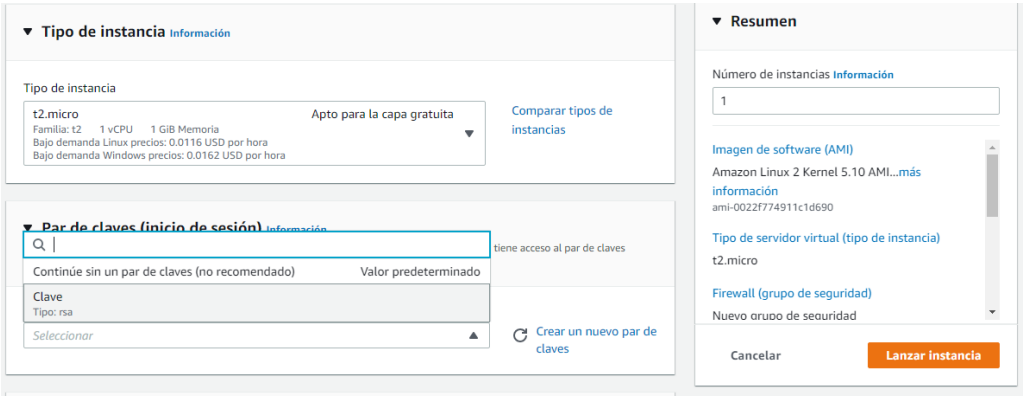


Figura 33. Creación de la instancia (Parte 5).

- Los grupos de seguridad actúan como firewall para las instancias asociadas al controlar el tráfico entrante y saliente en el ámbito de la instancia.
- Se debe agregar reglas a un grupo de seguridad que le permita conectarse a la instancia desde su dirección IP mediante SSH.
- También se pueden añadir reglas que permitan HTTP de entrada y salida y acceso HTTPS desde cualquier lugar.
- Si se desea lanzar instancias en varias regiones, se debe crear un grupo de seguridad por región.

Configuración de la instancia *AWS*

En configuración de la red se muestran los botones habilitados y deshabilitados.

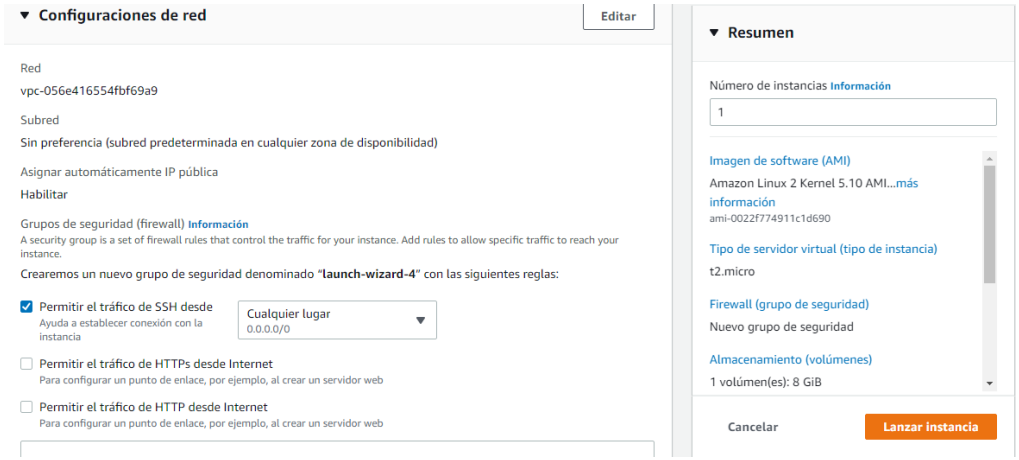


Figura 34. Configuración de la instancia (Parte 1).

Habilitar todos para permitir el tráfico en el servidor.

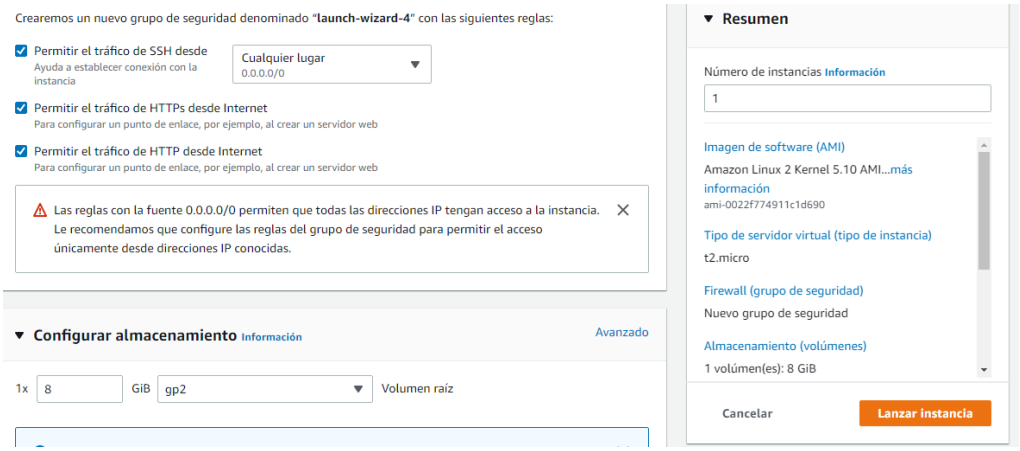


Figura 35. Configuración de la instancia (Parte 2).

Se muestra su almacenamiento de 8gb.

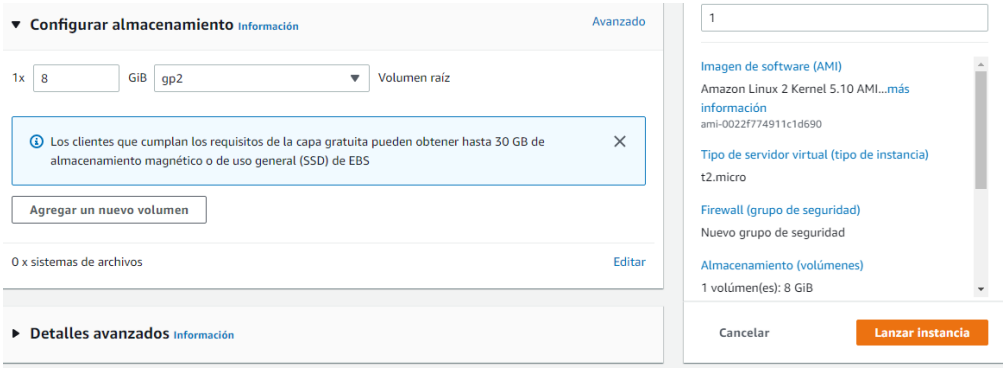


Figura 36. Configuración de la instancia (Parte 3).

Se termina la configuración para lanzar la instancia.

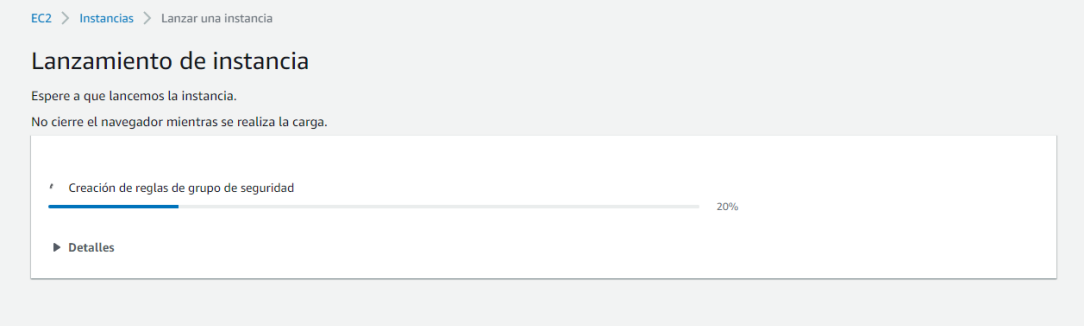


Figura 37. Configuración de la instancia (Parte 4).

Lanzamiento de la instancia

Se lanza la instancia con la configuración dada anteriormente.

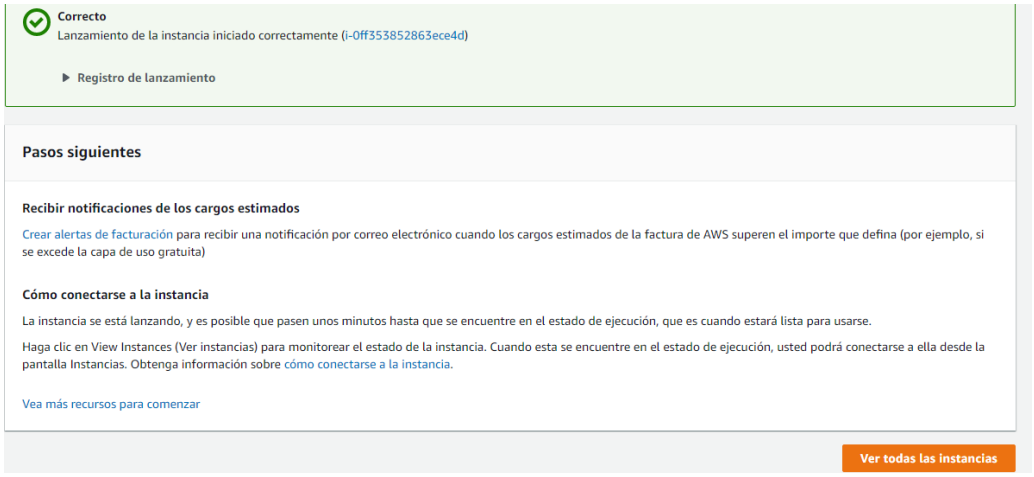


Figura 38. Lanzamiento de la instancia (Parte 1).

Se muestra la instancia creada y en ejecución.

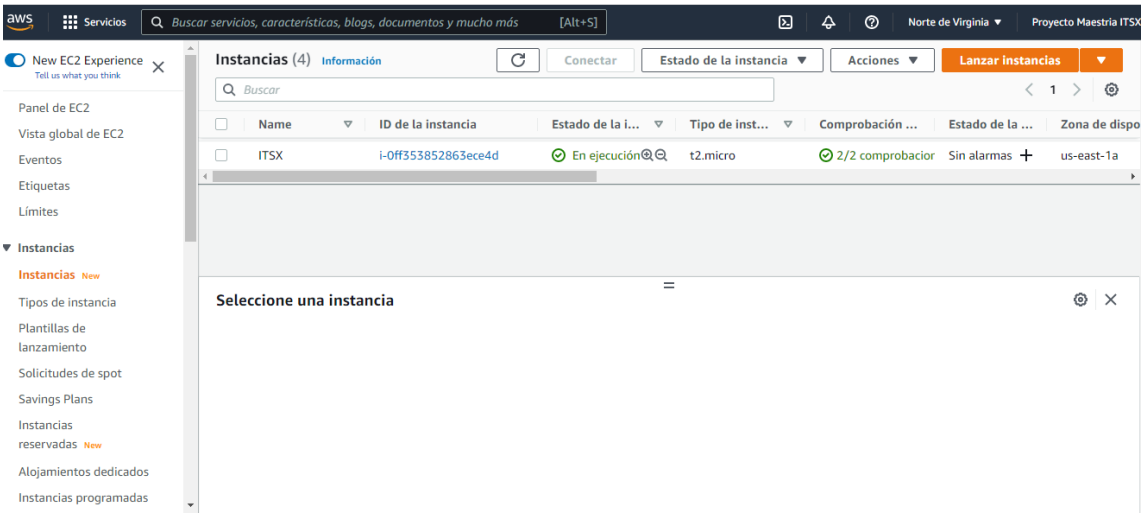


Figura 39. Lanzamiento de la instancia (Parte 2).

Se selecciona la instancia para conectarse en ésta.

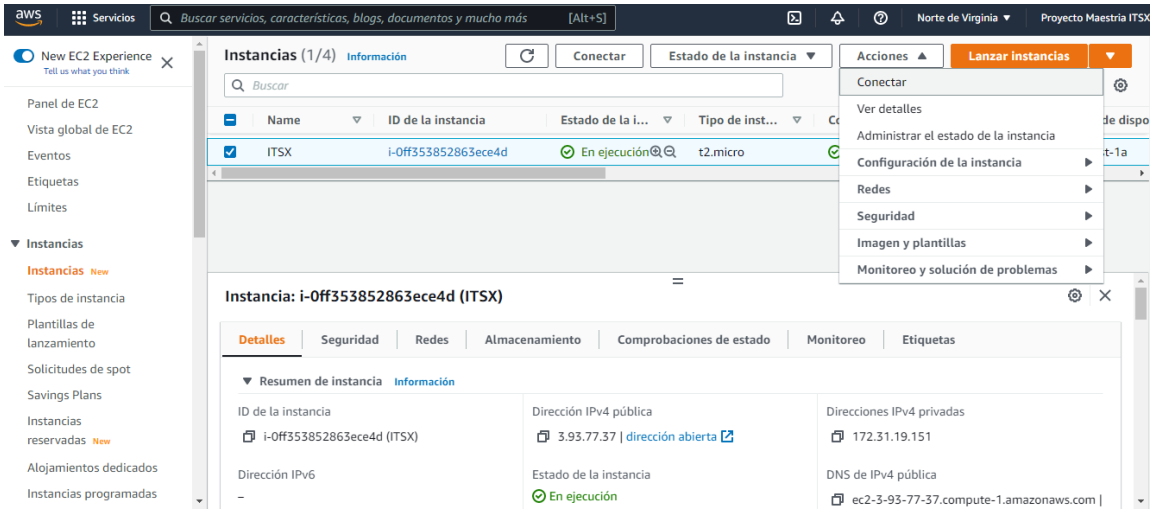


Figura 40. Conexión a la instancia (Parte 1).

Se da clic en la opción conectar.

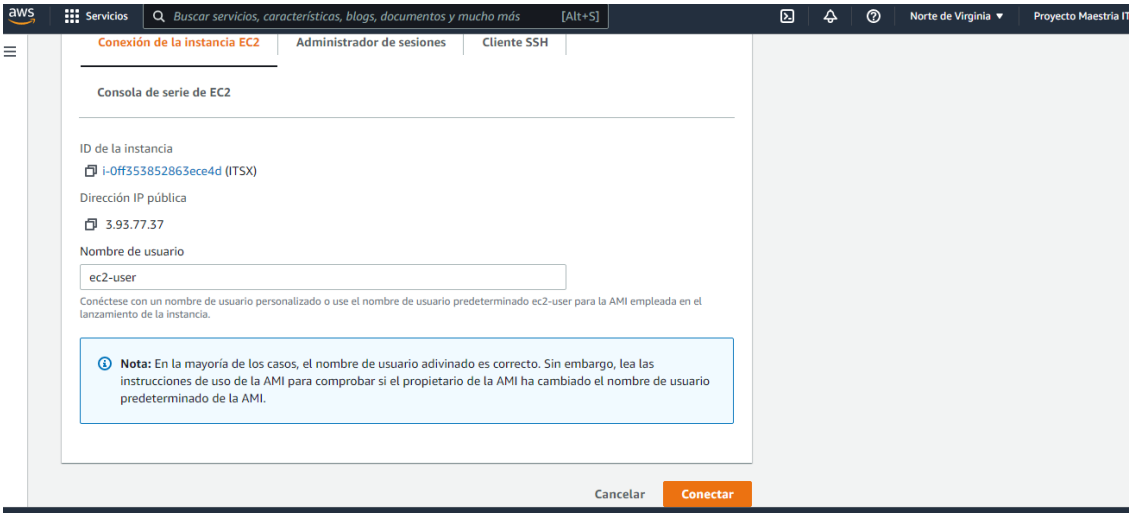


Figura 41. Conexión a la instancia (Parte 2).

Se tiene lista la instancia para trabajar en ella.

Figura 42. Conexión a la instancia (Parte 3).

Características de la instancia

A continuación se muestra las caracteristicas de la máquina virtual.

Figura 43. Características de la instancia (Parte 1).

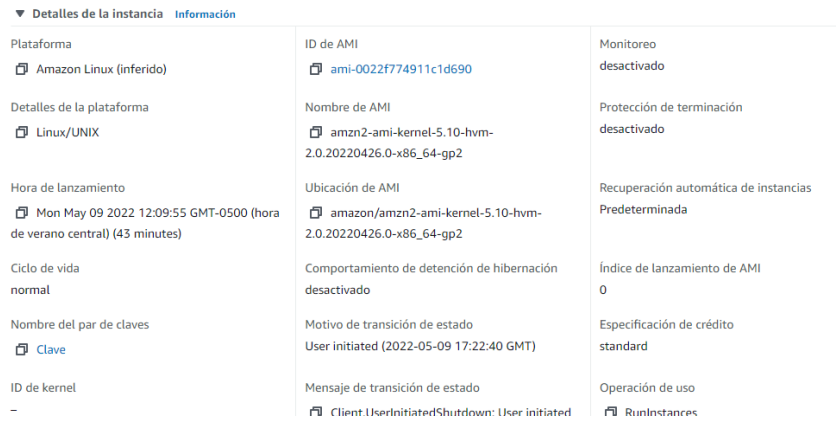


Figura 44. Características de la instancia (Parte 2).

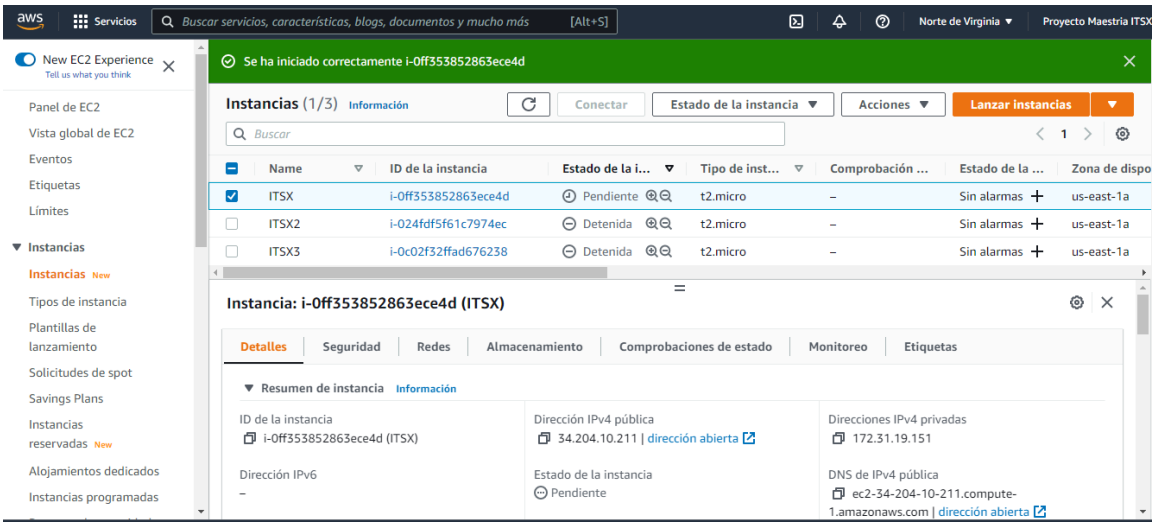


Figura 45. Características de la instancia (Parte 3).

Conexión a la instancia

Como se muestra la conexión a la instancia se muestra ID, sin IP pública y con el nombre del usuario llamado ec2-user.

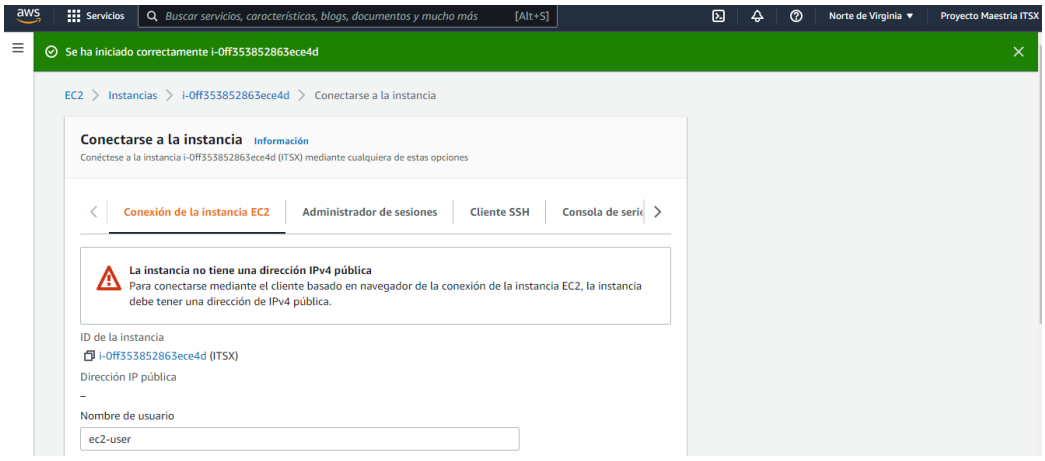


Figura 46. Conexión a la instancia (Parte 1).

Se conecta y se asigna una IP en este caso pública.

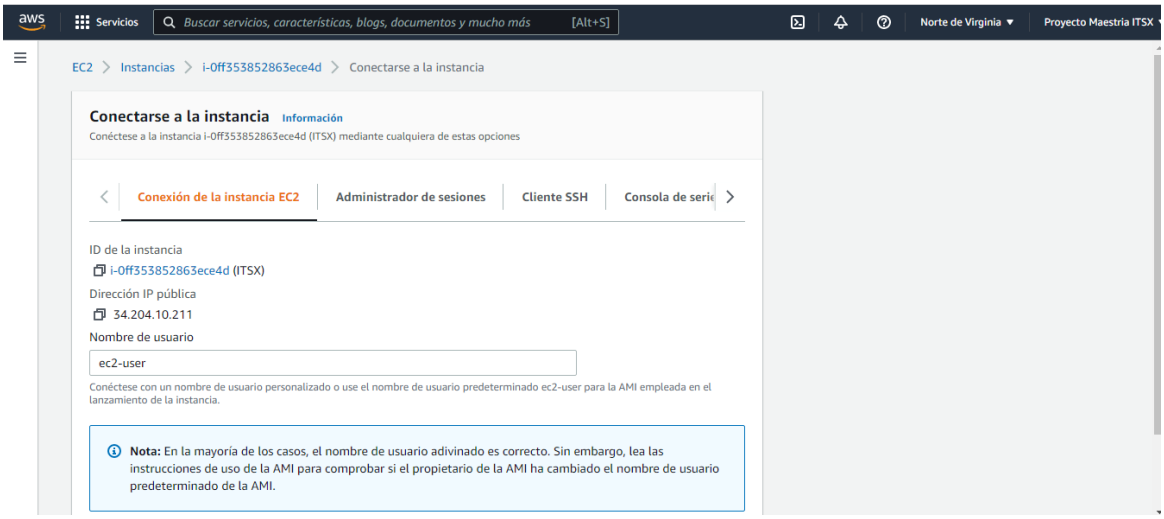


Figura 47. Conexión a la instancia (Parte 2).

Cliente SSH se muestra el ID de la instancia para que con clave.pem se conecte con PuTTY mediante la consola.

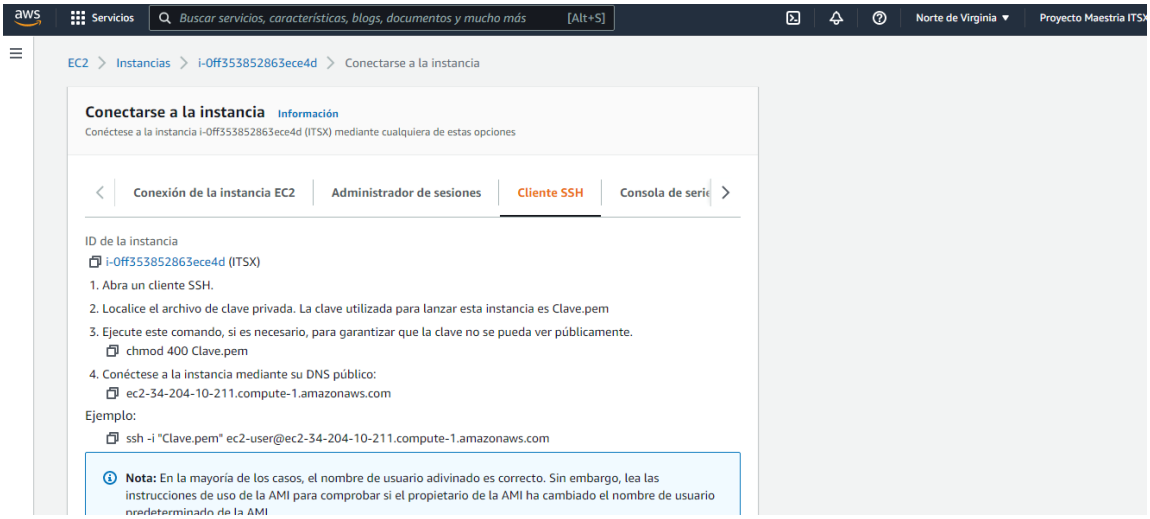


Figura 48. Conexión a la instancia (Parte 3).

Consola donde se muestra el puerto serie, el cual es ty50.

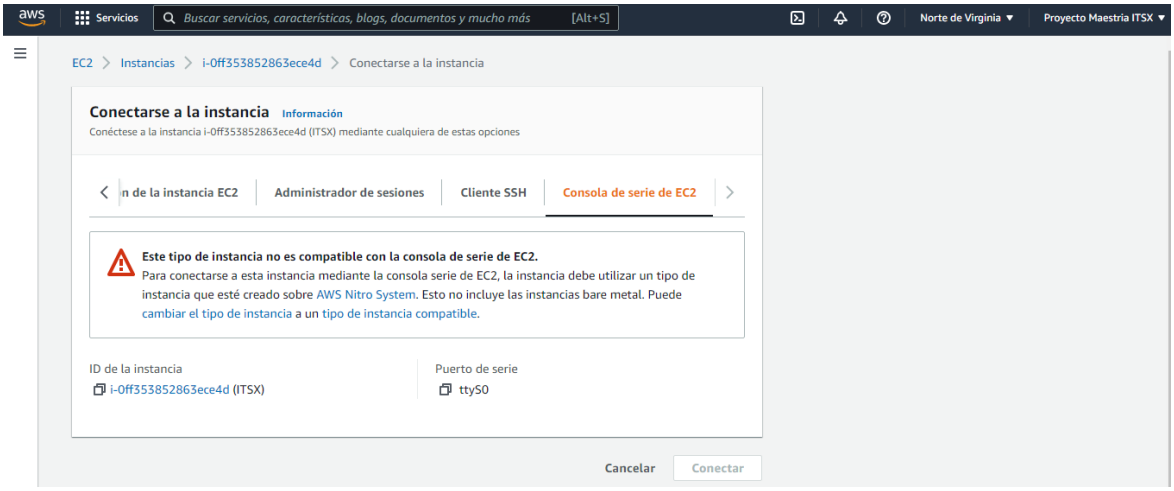


Figura 49. Conexión a la instancia (Parte 4).

Se conecta y la instancia *Amazon Linux* comienza para instalar las bibliotecas del proyecto.

Instalando bibliotecas

- Python3 –versión. Muestra la versión que se está trabajando.
- Pip3 –versión. Muestra donde se encuentra este paquete.

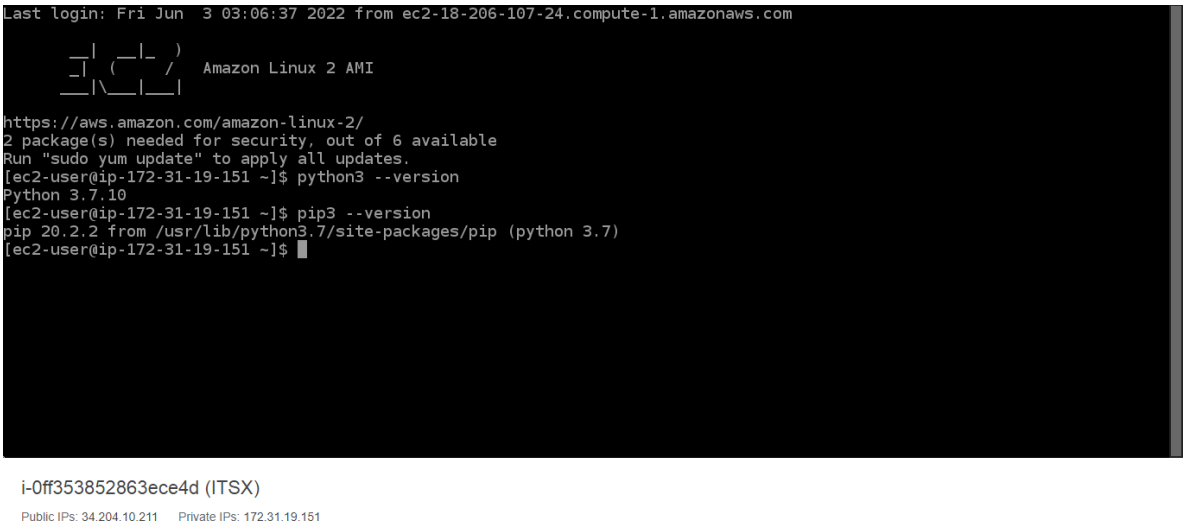


Figura 50. Instalando bibliotecas (Parte 1).

Ya que se vio la versión de *Python* y su ubicación a continuación se crea una carpeta llamada proyecto la cual tendrá el código del proyecto para poder ejecutarlo en la instancia.

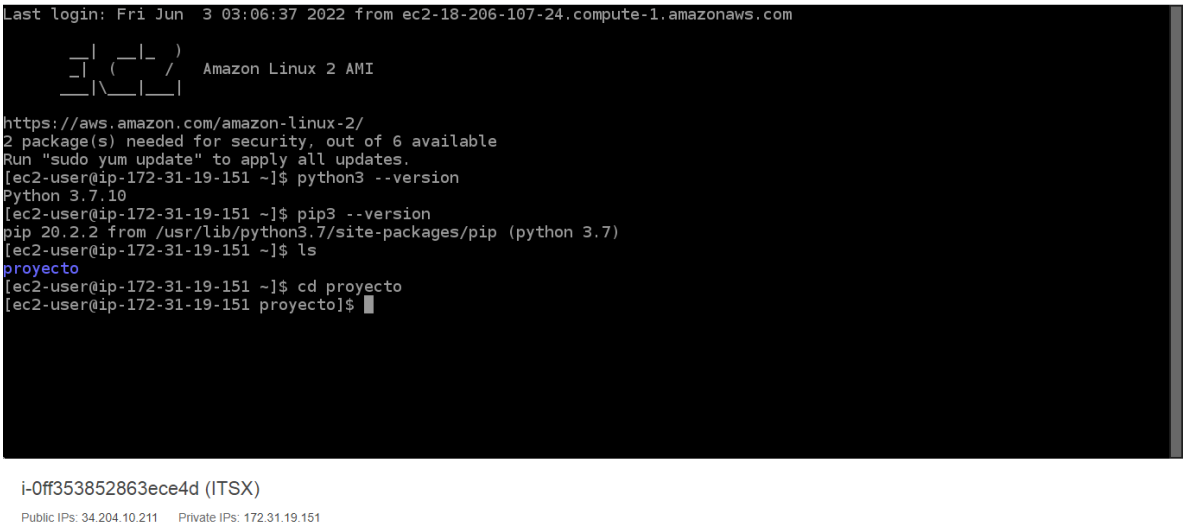


Figura 51. Instalando bibliotecas (Parte 2).

Conectar PuTTY: Para conectarse se debe de descargar la aplicación PuTTY el cual es un emulador de terminal gratuito que admite varios protocolos de red tal como SSH. Esto te permite correr comandos UNIX en tu servidor el cual no está disponible cuando te conectas usando un cliente FTP.

Una vez lanzada la instancia, pueden transcurrir unos minutos hasta que esté lista para conectarse. Verifique que su instancia ha pasado las comprobaciones de estado.

Para encontrar el nombre de DNS público o la dirección IP de la instancia y el nombre de usuario que debería utilizar para conectarse a la instancia.

Convertir la clave .pem privada en .ppk con PuTTYgen: Para el par de claves especificado al lanzar la instancia, crear la clave privada en formato .pem, debe convertirse en un archivo .ppk para usar con PuTTY. Buscar el archivo .pem privado.

Convertir la clave privada utilizando PuTTYgen: PuTTY no admite de forma nativa el formato PEM para claves SSH. PuTTY proporciona una herramienta llamada PuTTYgen, la cual convierte claves al formato requerido PPK para PuTTY.

Debe convertir su clave privada (archivo .pem) a este formato (archivo .ppk) como se indica a continuación para conectarse a la instancia mediante PuTTY.

Convertir una clave .pem privada al formato .ppk

En el menú Start (Inicio), elija All Programs (Todos los programas), PuTTY, PuTTYgen.

En Type of key to generate (Tipo de clave a generar), elegir RSA. Si la versión de PuTTYGen no incluye esta opción, elija SSH-2 RSA.

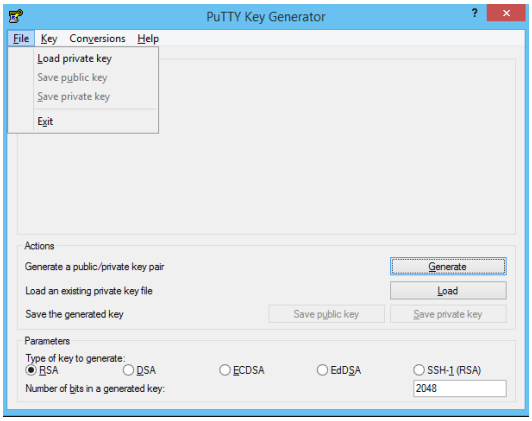


Figura 52. Generar la clave (Parte 1).

Elegir Load (Cargar). De forma predeterminada, PuTTYgen muestra solo archivos con la extensión .ppk. Para localizar el archivo .pem, seleccionar la opción de mostrar todos los tipos de archivo.

Figura 53. Generar la clave (Parte 2).

Seleccionar el archivo .pem para el par de claves que se especificó cuándo se lanzó la instancia y, a continuación, elija Open (Abrir). PuTTYgen muestra un aviso de que el archivo .pem se ha importado correctamente. Seleccionar OK.

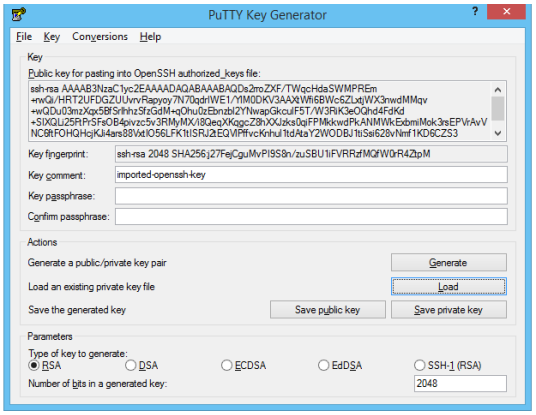


Figura 54. Generar la clave (Parte 3).

Elegir Save private key (Guardar la clave privada) para guardar la clave en formato que PuTTY pueda utilizar. PuTTYgen mostrará una advertencia acerca de guardar la clave sin una frase de contraseña. Elegir Yes (Sí).

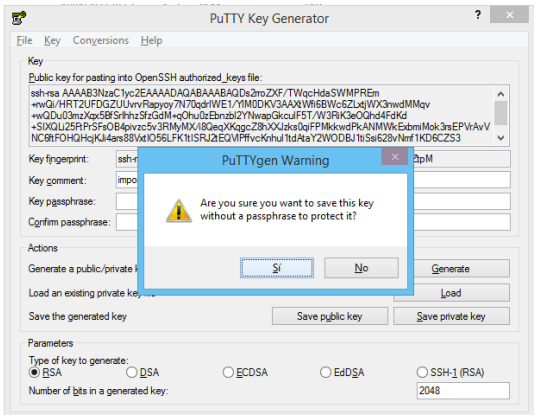


Figura 55. Generar la clave (Parte 4).

Especificar el mismo nombre para la clave que se utilizó para el par de claves (por ejemplo my-key-pair) y elija Save (Guardar). PuTTY añade la extensión de archivo .ppk automáticamente.

La clave privada está ahora en el formato correcto para su uso con PuTTY. Ya puede conectarse a la instancia mediante el cliente SSH de PuTTY.

Para conectar a la instancia de Linux mediante PuTTY, usar el siguiente procedimiento. Se necesita el archivo .ppk que creó para la clave privada.

Conectar a la instancia mediante PuTTY

Iniciar PuTTY (en el menú Inicio, elegir Todos los programas, PuTTY, PuTTY).
En el panel Category (Categoría), elegir Session (Sesión) y rellenar los siguientes campos:

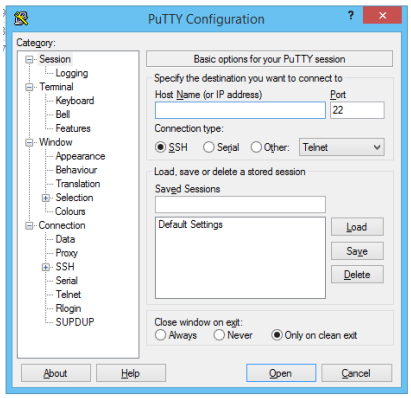


Figura 56. Iniciar la conexión con la clave (Parte 1).

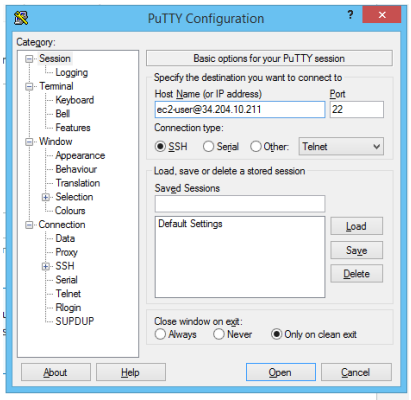


Figura 57. Iniciar la conexión con la clave (Parte 2).

En el cuadro Host Name (Nombre de host) seguir uno de estos procedimientos:

- (DNS público) Para conectarse utilizando el nombre DNS público de la instancia, escriba *my-instance-user-name@my-instance-public-dns-name*.
- (IPv6) Como opción, si la instancia tiene una dirección IPv6, para conectarse utilizando esta dirección IPv6, escriba *my-instance-user-name@my-instance-IPv6-address*.

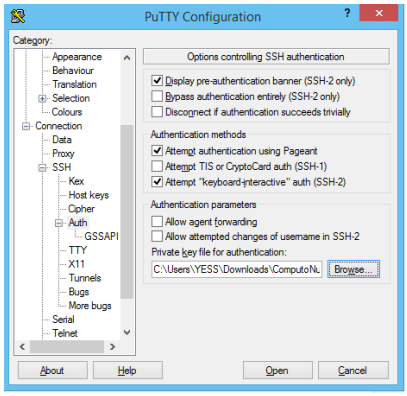


Figura 58. Iniciar la conexión con la clave (Parte 3).

El valor del Port (Puerto) es 22.

Implementación en la máquina virtual

A continuación se muestra la implementación de la máquina virtual y como se procede a hacer la conexión, se selecciona *Connection type* (Tipo de conexión), seleccionar *SSH.*

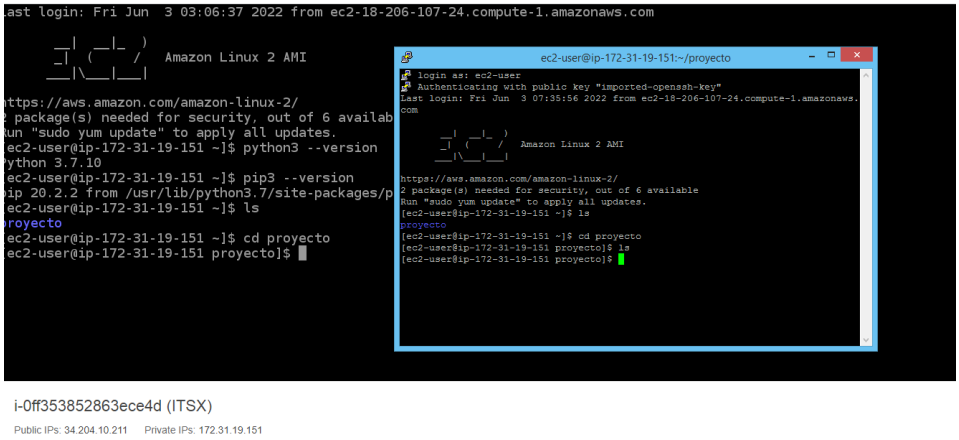


Figura 59. Conexión al tipo de instancia.

A continuación se procede a seleccionar la dirección ip para la máquina virtual en donde se tiene el proyecto con el script para ejecutar.

Dirección IP: pscp -i C:\Users\YESS\Downloads\ComputoNube\Claves AWS\convertirPutty\clave.ppk C:\Users\YESS\Downloads\Proyecto Mineria\pdftotextplumber.py ec2-user@ec2-34-204-10-211.compute-1.amazonaws.com:/home/proyecto/pdftotextplumber.py

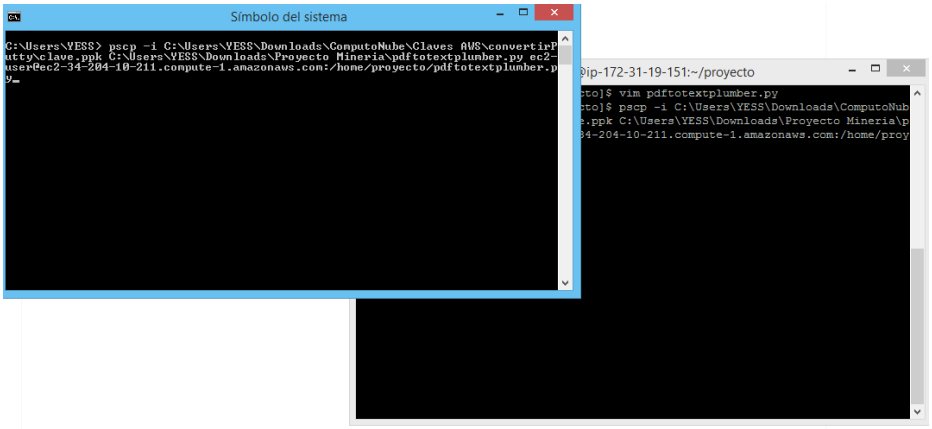


Figura 60. Ejecución del script en Python.

Se ejecuta el *script* creado en *Python*, el cual se carga en la instancia.

```
[ec2-user@ip-172-31-19-151 proyecto]$ vim pdftotextplumber.py
[ec2-user@ip-172-31-19-151 proyecto]$ ls
pdftotextplumber.py
[ec2-user@ip-172-31-19-151 proyecto]$
```

Figura 61. Script cargado a la instancia.

Se muestra el *script* el cual está hecho en *Python* con la biblioteca Pdfplumber para el preprocesamiento de textos. Para cada sección del código se describen los detalles de implementación.

```
    line_fi = file_name + " " + str(num_paginas) + " " + str(num_words) + " " +
str(total_time) + " " + str(psutil.virtual_memory()[2]) + " " + str(psutil.virtu
al_memory().total)

    with open('ley.csv', 'a+') as f2:
        wr = csv.writer(f2)
        wr.writerow(line_fi.split(' '))

# Aqui empieza a ejecutarse el programa

# crea archivo csv y agrega encabezado
with open('ley.csv', 'w', newline='') as headercsv:
    writer = csv.DictWriter(headercsv, fieldnames = ["filename", "page_number",
"num_words", "total time", "memoryporcent", "memorytotal"])
    writer.writeheader()


d = r"C:\Users\YESS\Downloads\Proyecto Mineria\Leyes"
for path in os.listdir(d):
    full_path = os.path.join(d, path)
    if os.path.isfile(full_path):
        open_file(full_path)
[ec2-user@ip-172-31-19-151 proyecto]$ clear
[ec2-user@ip-172-31-19-151 proyecto]$ vim pdftotextplumber.py
        msg = "El archivo " + filename2 + " no existe"
        print(msg)
    else:
        words = contents.split()
        num_words = len(words)
        print("El archivo " + filename2 + " contiene: " + str(num_words) + " palabras")

    end = timeit.timeit()
    total_time= end - start
    # print (os.path.splitext(string">"file_path")[0])

#  Fuente: https://www.iteramos.com/pregunta/5172/como-obtener-el-nombre-del-archivo-sin-la-extension-de-una-ruta-en-python
    print('Nombre del archivo: ', file_path)
    print('Numero de paginas: ', num_paginas)
    print('Numero de paginas: ', num_words)
    print('Tiempo de ejecucion: ', end - start)
    print('Memoria RAM del Proceso: ', psutil.virtual_memory()[2])
    print('Memoria RAM TOTAL: ', psutil.virtual_memory().total)
```

Figura 62. Script en Python (Parte 1).

```
      msg = "El archivo " + filename2 + " no existe"
      print(msg)
  else:
      words = contents.split()
      num_words = len(words)
      print("El archivo " + filename2 + " contiene: " + str(num_words) + " palabras")

  end = timeit.timeit()
  total_time= end - start
  # print (os.path.splitext(string">"file_path")[0])

 Fuente: https://www.iteramos.com/pregunta/5172/como-obtener-el-nombre-del-archivo-sin-la-extension-de-una-ruta-en-python
  print('Nombre del archivo: ', file_path)
  print('Numero de paginas: ', num_paginas)
  print('Numero de paginas: ', num_words)
  print('Tiempo de ejecucion: ', end - start)
  print('Memoria RAM del Proceso: ', psutil.virtual_memory()[2])
  print('Memoria RAM TOTAL: ', psutil.virtual_memory().total)

  line_fi = file_name + " " + str(num_paginas) + " " + str(num_words) + " " + str(total_time) + " " + str(psutil.virtual_memory()[2]) + "
().total)

  with open('ley.csv', 'a+') as f2:
      wr = csv.writer(f2)
      wr.writerow(line_fi.split(' '))

Aqui empieza a ejecutarse el programa

crea archivo csv y agrega encabezado
ith open('ley.csv', 'w', newline='') as headercsv:
  writer = csv.DictWriter(headercsv, fieldnames = ["filename", "page_number", "num_words", "total time", "memoryporcent", "memorytotal"])
  writer.writeheader()


 = r"C:\Users\YESS\Downloads\Proyecto Mineria\Leyes"
or path in os.listdir(d):
  full_path = os.path.join(d, path)
  if os.path.isfile(full_path):
      open_file(full_path)
```

Figura 63. Script en Python (Parte 2).

```
[ec2-user@ip-172-31-19-151 proyecto]$ vim pdftotextplumber.py
[ec2-user@ip-172-31-19-151 proyecto]$ ls
pdftotextplumber.py
[ec2-user@ip-172-31-19-151 proyecto]$ note pdftotextplumber.py
-bash: note: command not found
[ec2-user@ip-172-31-19-151 proyecto]$ cat pdftotextplumber.py
import pdfplumber
from nltk.tokenize import word_tokenize
from nltk import sent_tokenize
import os
import timeit
import psutil
import string
import re
import csv

start = timeit.timeit()
# file_path = 'Ley General del Equilibrio Ecologico y la Proteccion al Ambiente.
pdf'
# file_path = 'Ley de Aguas Nacionales.pdf'
# file_path = 'Ley de Desarrollo Rural Sustentable.pdf'
# file_path = 'Ley General de Pesca y Acuacultura Sustentable.pdf'
# file_path = 'Ley General de Vida Silvestre.pdf'
# file_path = 'Ley General de Desarrollo Forestal Sustentable.pdf'
# file_path = 'Ley General Para la Prevención y Gestión Integral de Residuos.pdf
'
# file_path = 'Ley Federal de Bioseguridad de Organismos Genéticamente Modificad
os.pdf'
# file_path = 'Ley General De Cambio Climático.pdf'

# Funcion open_file que realiza el proceso de conversion de pdf a text y separac
ion de palabras
def open_file(file_path):
    file_name = os.path.basename(file_path)
    pdf = pdfplumber.open(file_path)

    with open('text.txt', 'w') as f:
        for page_num in pdf.pages:
            # print('Page: {0}'.format(page_num))
            #pageObj = pdf.pages[0]
            num_paginas = page_num
            try:
                txt = page_num.extract_text()
                # print(''.center(100, '-'))
```

Figura 64. Script en Python (Parte 3).

A continuación, se importan las bibliotecas *nltk* para que el programa se ejecute adecuadamente con *pip install nltk*.

```
[ec2-user@ip-172-31-19-151 proyecto]$ pip install nltk
-bash: pip: command not found
[ec2-user@ip-172-31-19-151 proyecto]$ pip3 install nltk
Defaulting to user installation because normal site-packages is not writeable
Collecting nltk
  Downloading nltk-3.7-py3-none-any.whl (1.5 MB)
     |████████████████████████████████| 1.5 MB 28.8 MB/s
Collecting regex>=2021.8.3
  Downloading regex-2022.6.2-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (749 kB)
     |████████████████████████████████| 749 kB 34.9 MB/s
Collecting tqdm
  Downloading tqdm-4.64.0-py2.py3-none-any.whl (78 kB)
     |████████████████████████████████| 78 kB 10.7 MB/s
Collecting joblib
  Downloading joblib-1.1.0-py2.py3-none-any.whl (306 kB)
     |████████████████████████████████| 306 kB 35.6 MB/s
Requirement already satisfied: click in /home/ec2-user/.local/lib/python3.7/site-packages (from nltk) (8.1.3)
Requirement already satisfied: importlib-metadata; python_version < "3.8" in /home/ec2-user/.local/lib/python3.7/site-packages (from click->nltk) (4.11.4)
Requirement already satisfied: typing-extensions>=3.6.4; python_version < "3.8" in /home/ec2-user/.local/lib/python3.7/site-packages (from importlib-metadata; python_v
rsion < "3.8"->click->nltk) (4.2.0)
Requirement already satisfied: zipp>=0.5 in /home/ec2-user/.local/lib/python3.7/site-packages (from importlib-metadata; python_version < "3.8"->click->nltk) (3.8.0)
Installing collected packages: regex, tqdm, joblib, nltk
Successfully installed joblib-1.1.0 nltk-3.7 regex-2022.6.2 tqdm-4.64.0
[ec2-user@ip-172-31-19-151 proyecto]$
```

Figura 65. Importación de bibliotecas.

Ya que se tiene instalado el *script*, los comandos para ejecutar son:

- ec2-34-204-10-211.compute-1.amazonaws.com
- ec2-user@ec2-34-204-10-211.compute-1.amazonaws.com
- ec2-user
- cd proyecto/
- ls
- python3 pdftotextpluber.py

Se muestra que el *script* se ejecutó en la instancia con éxito.

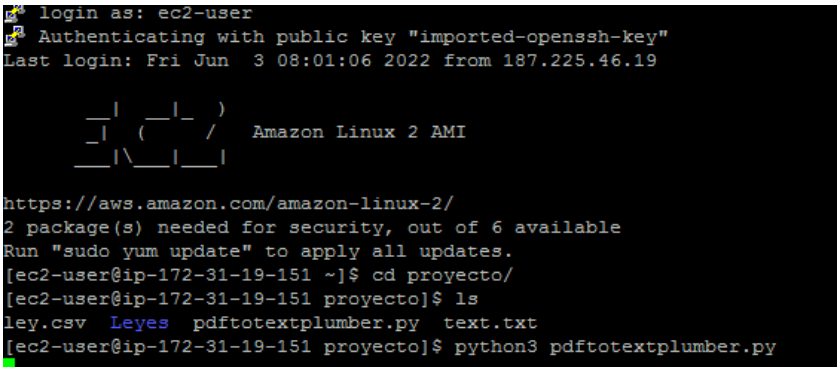


Figura 66. Script ejecutado con éxito (Parte 1).

Como se muestra en la Figura 29, una instancia es un servidor virtual en la nube de *AWS*. Con *Amazon EC2* se puede instalar y configurar el sistema operativo y las aplicaciones que se ejecutan en la instancia.

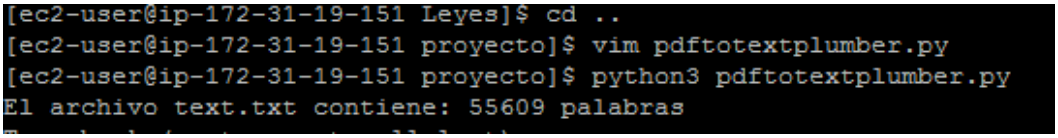


Figura 67. Script ejecutado con éxito (Parte 2).

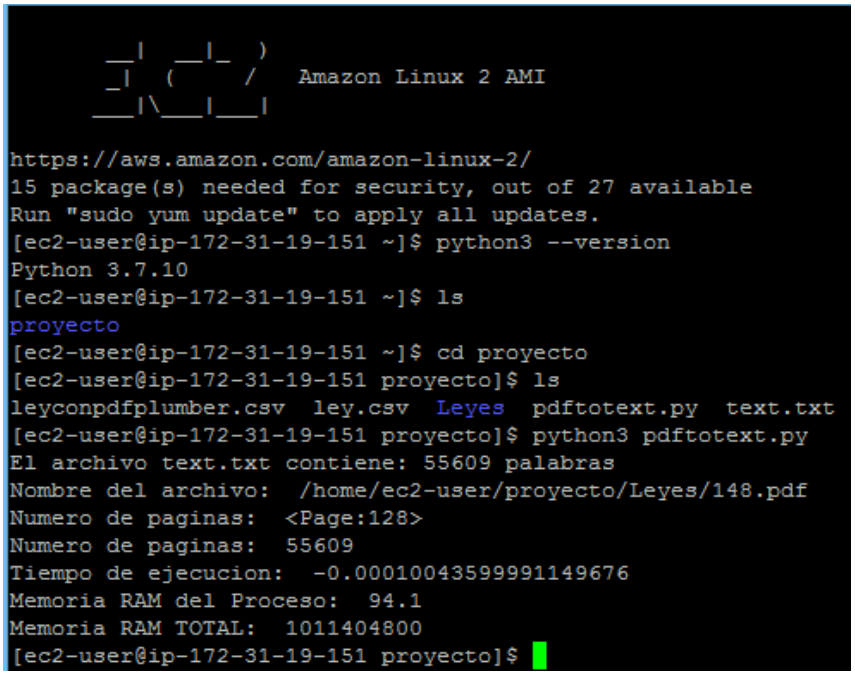


Figura 68. Resultados en la instancia.

Al conectarse a la instancia se debe especificar la clave privada del par de claves que especificó cuando lanzó la instancia, ya que no todas tienen las mejores características en la capa gratuita. Por razones de seguridad y monitoreo, se seleccionó esta instancia, al igual que PuTTY el cual es un emulador para el manejo de esta instancia.