EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

Tecnológico Nacional de México

Centro Nacional de Investigación

y Desarrollo Tecnológico

Tesis de Maestría

Desarrollo de un sistema de Web Scraping para la obtención de datos en entornos Big Data

Presentado por

Ing. Rogelio Daniel Mijangos Espinosa

Como requisito para la obtención del grado de

Maestro en Ciencias de la Computación

Director de tesis

Dr. Hugo Estrada Esquivel

Codirector de tesis

Dra. Alicia Martínez Rebollar

Cuernavaca, Morelos, México. 10 de febrero del 2023

cenidet
Centro Nacional de Investigación y Desarrollo Tecnológico

EDUCACIÓN

Centro Nacional de Investigación y Desarrollo Tecnológico

Cuernavaca, Morelos, 02/febrero/2023
No. de Oficio: DCC/037/2023
Asunto: Aceptación de documento de tesis
CENIDET-AC-004-M14-OFICIO

JUAN GABRIEL GONZÁLEZ SERNA
JEFE DEL DEPARTAMENTO DE CIENCIAS COMPUTACIONALES
PRESENTE

Por este conducto, los integrantes de Comité Tutorial de ROGELIO DANIEL MOANCOS ESPINOSA con número de control M20CE066, de la Maestría en Ciencias de la Computación, le informamos que hemos revisado el trabajo de tesis de grado titulado ***"DESARROLLO DE UN SISTEMA DE WEB SCRAPING PARA LA OBTENCIÓN DE DATOS EN ENTORNOS BIG DATA"*** y hemos encontrado que se han atendido todas las observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.

HUGO ESTRADA ESQUIVEL
Director de tesis

ALICIA MARTÍNEZ REBOLLAR
Codirectora de tesis

JOAQUÍN PÉREZ ORTEGA
Revisor 1

MARÍA YASMÍN HERNÁNDEZ PÉREZ
Revisor 2

EDUCACIÓN
09 FEB 2023
RECIBIDO

c.c.p. Archivo

cenidet

2023 Francisco VILLA

EDUCACIÓN

Cuernavaca, Mor., 09/febrero/2023
No. De Oficio: SAC/046/2023
Asunto: Autorización para presentar examen de grado

ROGELIO DANIEL MIJANGOS ESPINOSA
CANDIDATO AL GRADO DE MAESTRO EN CIENCIAS DE LA COMPUTACIÓN
P R E S E N T E

Me es grato comunicarle que una vez cubiertos todos los requisitos necesarios para presentar el examen de grado de Maestra en Ciencias de la Computación, con la tesis titulada **"DESARROLLO DE UN SISTEMA DE WEB SCRAPING PARA LA OBTENCIÓN DE DATOS EN ENTORNOS BIG DATA"**, dirigida por Dr. Hugo Estrada Esquivel, **SE AUTORIZA** la presentacion del mismo el día 13 de febrero del 2023, a las 10:00 horas.

Aprovecho la ocasion para desearle el mejor de los éxitos en su examen, así como en su vida profesional, y agradecerle la confianza depositada en nuestra institución para la realización de sus estudios.

A T E N T A M E N T E

EDUCACIÓN
ADMINISTRACIÓN ACADÉMICA

CARLOS MANUEL ASTORGA ZARAGOZA
SUBDIRECTOR ACADÉMICO

C. c. p. Departamento de Ciencias Computacionales
Departamento de Servicios Escolares
CMAZ/BHJL

cenidet

2023 Felicidad VIDA

Dedicatoria

A mis padres Juan Daniel Mijangos García y Rita Herminia Espinosa Quintas por mostrarme que el esfuerzo, la dedicación y las ganas de salir adelante nos permiten ser mejores personas con grandes recompensas, **ellos serán siempre mi ejemplo a seguir**.

A mi esposa Sua Abigail Munguia Reyes por brindarme todo su apoyo y ser ese soporte que me permite avanzar día a día con pasos firmes, sabiendo que tengo a una mujer fuerte, inteligente y hermosa que saca lo mejor de mí a cada momento.

A mi hija Leah Mikela Mijangos Munguia por ser mi motivación y mi mayor orgullo, **Te amo hija.**

Agradecimientos

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico que me brindó para realizar mis estudios de maestría.

Al Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) por brindarme la oportunidad de continuar mi preparación académica en el programa de maestría en ciencias de la computación.

A mi director de tesis, el Dr. Hugo Estrada Esquivel y a mi codirectora, la Dra. Alicia Martínez Rebollar, por todos los consejos, el tiempo brindado, los regaños, la paciencia, los conocimientos y el interés por el trabajo realizado en esta investigación.

A mis revisores, el Dr. Joaquín Pérez Ortega y la Dra. María Yasmín Hernández Pérez por los comentarios asertivos y aportaciones que fortalecieron mi trabajo de investigación.

A mis amigos y compañeros de maestría y doctorado que me brindaron su apoyo, fortaleciendo mis conocimientos.

Resumen

Los bancos de datos Web son entornos que ofrecen acceso a grandes cantidades de información, la cual es generada a cada momento en internet. Estos bancos de datos Web se nutren de páginas Web, bases de datos, datasets, sensores, redes sociales y cualquier aparato electrónico con conexión a internet. La información generada por estas fuentes de información es de gran importancia para las áreas académicas, laborales y personales, ya que genera información que es actualizada en forma mucho más dinámica que las fuentes convencionales de consulta bibliográfica, como son los libros de consulta.

Sin embargo, el crecimiento exponencial y descontrolado de información en la Web complican las actividades de búsqueda, recolección y preprocesamiento de la información. Esta tendencia ha ocasionado la creación de un enorme volumen de datos tanto estructurados como no estructurados. Estos entornos requieren de herramientas de Big Data para analizar e interpretar estos conjuntos de datos. Los sistemas de *Web Scraping* (Raspado Web) son programas informáticos que simulan la navegación de una persona dentro de un sitio Web y permiten realizar las tareas de búsqueda, recolección y procesamiento de información contenida en internet de forma automática, lo cual permite reducir el tiempo y esfuerzo requerido para obtener información de un tipo específico.

En este trabajo de investigación se presenta la propuesta de un sistema de obtención de datos basado en técnicas de *Web Scraping*. Este sistema permite la búsqueda de información en páginas Web de un tema específico, por ejemplo, información relacionada con COVID. El sistema permite que la obtención de una determinada fuente de información se realice de forma automática con la periodicidad especificada por el usuario. La búsqueda de la información es configurada por el usuario por lo que el sistema requerirá parámetros de configuración para la realización de la búsqueda, recolección y almacenamiento de los datos. Como resultado, es posible automatizar el proceso de obtención y almacenamiento de datos que se desean obtener de forma recurrente a partir de la Web. La solución propuesta en esta tesis permite reducir el tiempo que un investigador dedica a obtener información recurrente para un tópico específico. Esta solución es de especial utilidad en entornos Big Data, donde se requiere la recuperación de grandes volúmenes de información de múltiples sitios, lo cuales pueden ser actualizados continuamente.

Abstract

Web databanks are environments that offer access to large amounts of information, which is generated all the time on the Internet. These Web databanks are fed by Web pages, databases, datasets, sensors, social networks and any electronic device with an Internet connection. The information generated by these information sources is of great importance for academic, work and personal areas, since it generates information that is updated much more dynamically than conventional sources of bibliographic consultation, such as reference books.

However, the exponential and uncontrolled growth of information on the Web complicates information search, collection and preprocessing activities. This trend has led to the creation of an enormous volume of both structured and unstructured data. These environments require Big Data tools to analyze and interpret these data sets. Web Scraping systems are computer programs that simulate the navigation of a person within a Web site and allow to perform the tasks of searching, collecting and processing information contained in the Internet automatically, which reduces the time and effort required to obtain information of a specific type.

This research work presents the proposal of a data collection system based on Web Scraping techniques. This system allows the search of information in Web pages of a specific topic, for example, information related to COVID. The system allows the retrieval of a specific source of information to be performed automatically with the periodicity specified by the user. The search for the information is configured by the user so the system will require configuration parameters for performing the search, collection and storage of the data. As a result, it is possible to automate the process of obtaining and storing data to be obtained on a recurring basis from the Web. The solution proposed in this thesis allows to reduce the time that a researcher spends to obtain recurring information for a specific topic. This solution is especially useful in Big Data environments, where the retrieval of multiple sites, which can be continuously updated, is required.

Contenido

Lista de Figuras

Lista de Tablas

Capítulo 1
Introducción

1.1. Introducción

En la actualidad, la Web se ha convertido en una fuente de información muy importante, no sólo para fines de ocio sino para fines de investigación. Diariamente se produce una cantidad impresionante de información sobre un tópico específico, tanto en forma de comentarios en redes sociales, bases de datos, datasets, notas periodísticas, entre otras publicaciones. Esto ha ocasionado que múltiples grupos de investigación en el mundo se enfoquen en realizar búsquedas frecuentes de información para alimentar sus investigaciones con información oportuna.

Un punto de análisis en la información localizada en la Web es la calidad de los datos, ya que existe mucha información en la Web con información imprecisa o errónea que podría usarse para investigación que resulte también en resultados erróneos. Es por esto que los investigadores realizan esfuerzos para localizar aquellas fuentes de información confiable que puedan usar en forma recurrente. En este sentido han surgido los bancos de datos Web como entornos que ofrecen acceso a grandes cantidades de información la cual es generada por internet todos los días. Estos bancos de datos Web se nutren de páginas Web, bases de datos, datasets, sensores, redes sociales y cualquier aparato electrónico con conexión a internet. Los bancos de datos pueden ser de un volumen tan grande, tanto en datos estructurados como no estructurados, que requieren de técnicas de Big Data para analizar e interpretar estos conjuntos de datos.

La tarea de localización de información fidedigna es punto central de muchos grupos de investigación, los cuales, una vez que han logrado encontrar fuentes de información confiable, tienen que realizar actividades de descarga de información en forma periódica. Esta tarea, en entornos altamente dinámicos puede volverse una tarea que consuma demasiado tiempo y que pueda ocasionar desfases entre los datos actualizados y publicados en la Web con los datos que han podido ser descargados en forma manual por los investigadores.

El uso de sistemas informáticos es una herramienta de apoyo para facilitar la localización, descarga y organización de información fidedigna de forma constante.

1.2. Planteamiento del problema

La información contenida en los bancos de datos Web es utilizada como herramienta por empresas e instituciones. Información correcta empleada de forma adecuada puede reflejarse en reducción de costos en operaciones, implementación de medidas de prevención, creación de nuevos productos y servicios, y como un medio para incrementar las ganancias. Sin embargo, los cambios frecuentes en la información, el crecimiento exponencial de los datos

y la distribución de los mismos en la Web debe ser considerados para lograr prevenir los problemas de obtención y análisis de datos.

El crecimiento exponencial y descontrolado de información en la Web, aún en estos bancos de datos Web ha ocasionado que las actividades de búsqueda y recopilación de datos se convierta en una actividad que consume mucho tiempo y esfuerzo, sobre todo cuando se trata de buscar y recopilar información en la Web que se actualice de forma muy constante y que se encuentre distribuida en múltiples sitios Web. Por ejemplo, si un investigador desea recopilar la información que se produce diariamente sobre un tema como el COVID para fines estadísticos, podría utilizar gran parte del tiempo de investigación en buscar y descargar la información que se genera en múltiples sitios Web tanto de centros de investigación, entidades gubernamentales y universidades.

La búsqueda y extracción de información obtenida en forma manual puede provocar errores comunes como son datos nulos, datos incompletos o inconsistentes. Esto provoca que los análisis de datos posteriores serán poco confiables. En este sentido la búsqueda y extracción de información en forma continua se ha convertido en una actividad muy demandante, sobre todo en entornos de información donde la información se produce en forma muy rápida y donde es muy complicado llevar un mapeo entre la nueva información publicada y aquella que ya reside en las bases de datos de los investigadores.

En este sentido son necesarias herramientas que permitan a los usuarios enfrentar, en forma sistemática y automatizada, las tareas de búsqueda y recuperación de información, tomando en consideración que se cuenta con un banco específico de información de calidad que puedan ser consultadas de forma recurrente. Un sistema informático que realice las tareas de búsqueda y recolección de forma semiautomática permitirá obtener datos de calidad (Ram Sharan, Santosh, & Sadhu Ram, 2017). Un ejemplo actual donde se requiere datos de calidad y en tiempo real es la pandemia mundial provocada por el COVID19 el cual requiere de datos actualizados para un plan de respuesta basado en datos reales. La búsqueda de información para el COVID es un buen ejemplo de que son necesarios mecanismos de recuperación automática de información que pueda posteriormente ser utilizada en entornos Big Data.

Las técnicas de *Web Scraping* (Raspado Web) permiten la búsqueda, extracción y preprocesamiento de la información de manera automática. Esta tecnología crea peticiones a los sitios Web para obtener los datos contenidos en ella, almacenándolos en una nueva base de datos en la cual el usuario tiene el control de accesos (González Jaimez, 2015).

1.3. Justificación

Los entornos Big Data implican la recuperación de información de diversas fuentes, donde cada fuente puede ser actualizada en tiempos diferentes y con información representada en diferentes formatos. Esto implica un reto para los investigadores que requieren información oportuna basada en la Web, ya que implica la búsqueda y recuperación manual de

información a través de la interacción con el sitio Web donde se encuentran alojados los datos. En algunas ocasiones, la información solo puede ser recuperada con el llenado de formularios en el sitio Web y a través de acciones de selección para indicar el tipo de información requerida. Por ejemplo, para recuperar bases de datos de fuentes oficiales del COVID en nuestro país es necesario realizar procesos de selección de la localidad de la cual deseamos obtener datos, el año y mes que deseamos analizar y finalmente, el usuario debe realizar la acción de dar clic en los botones del sitio Web para poder tener acceso a una base de datos específica. Aunque esta puede parecer una actividad simple, puede resultar muy complicada cuando se desea obtener bases de datos de COVID de cada mes del año y para todas las localidades y, sobre todo, de información de múltiples sitios, donde cada uno de los sitios puede ser actualizado en periodos diferentes.

En estos entornos tan demandantes se requiere de herramientas que automaticen el proceso de obtención de información. En este sentido, las herramientas de *Web Scraping* resultan de gran utilidad para disminuir el tiempo y esfuerzo requerido para obtener información recurrente de múltiples sitios Web.

1.4. Objetivo general

Desarrollar un sistema de informático basado en técnicas de *Web Scraping* (Raspado Web) que permita obtener información de bancos de datos Web de forma automatizada.

1.5. Objetivos específicos

- Diseñar una propuesta de solución que permita la recuperación de grandes volúmenes de datos que pueden estar dispersos en diferentes sitios Web.
- Seleccionar los mejores métodos de rastreo Web que permitan la identificación y descarga de bases de datos de las fuentes de información.
- Diseñar y desarrollar los módulos de recuperación, transferencia y almacenamiento de información del sistema *Web scraping* para su uso en entornos Big Data.
- Realizar pruebas de funcionamiento al sistema *Web scraping* para un caso de estudio en específico y evaluar los resultados tomando como base los datos recuperados.

1.6. Estructura de la tesis

La estructura con la que se organiza este trabajo de organización es la siguiente:

- Capítulo 2 Marco teórico. Este capítulo define los conceptos principales utilizados en el desarrollo de este trabajo de investigación.

- Capítulo 3 Estado del arte. Este capítulo describe los trabajos de investigación relacionados con la creación y aplicación de técnicas de *Web Scraping* para la recolección de información Web.
- Capítulo 4 Metodología de solución. Este capítulo detalla las características y aplicación de las técnicas de *Web scraping* para la recolección de información Web.
- Capítulo 5 Experimentación y resultados. Este capítulo detalla los criterios, métricas y aplicaciones que fueron utilizados para la evaluación de la extracción de información Web para el caso de estudio del COVID en México.
- Capítulo 6 Conclusiones. Este capítulo presenta las conclusiones del trabajo de investigación realizado y muestra los trabajos a futuro.

Capítulo 2 Marco Teórico

Las técnicas de *Web Scraping* son relativamente recientes y han tomado mucha relevancia con la expansión acelerada de la información que se produce, almacena y expone en la Web. En este capítulo se describen algunos conceptos utilizados en la implementación de soluciones basadas en *Web Scraping*. Los conceptos se han agrupado en 3 categorías: Páginas Web y Bancos de Datos, Web Scraping y Métricas para medir la efectividad de las técnicas de Web Scraping. A continuación, se presenta cada una de estas categorías.

2.1. Páginas Web y Bancos de Datos

2.1.1. Información Web

La información Web es aquella que está elaborada en algún lenguaje de programación Web y cuya característica principal es estar conformada por documentos hipertextuales y multimedia. Las páginas Web contienen información en si misma o bien puede estar vinculada con otras páginas a través de hiperenlaces que completan la información generando así un espacio denominado sitio Web (Aldana, 2002).

La información almacenada en páginas Web puede ser de diferentes tipos como son textos, imágenes, audios o videos a los que se puede acceder utilizando un navegador Web. En su gran mayoría, esta información puede ser accedida y descargada en forma abierta.

2.1.2. Web estática

La Web estática es llamada así por sus páginas Web de contenido plano, es decir, que una vez cargada la página Web en el navegador el usuario observa el contenido de la página en su totalidad (González Jaimez, 2015).

Una página Web estática está compuesta por archivos HTML individuales por cada página, de esta manera, el navegador Web no necesita realizar solicitudes extra al servidor para poder mostrar el contenido completo de la página, ya que todos los elementos o información son cargados desde un inicio (González Moreno, 2001).

2.1.3. Web dinámica

Las páginas Web dinámicas son aquellas que tienen elementos en constante cambio o actualización en tiempo real. Este tipo de sitos Web permiten la interacción página -usuario con lo cual una acción genera nuevo contenido que ya se encuentra programado dentro de la página Web (González Jaimez, 2015).

Las páginas dinámicas, a diferencias de las estáticas, requieren de una programación compleja con elementos de programación avanzada. Este tipo de programación Web vuelve

complejo el manejo de su estructura, ya que la información contenida en la página se genera con las interacciones del usuario con el navegador (González Moreno, 2001).

Los sitios Web dinámicos basan su comportamiento y funcionalidad en dos tipos de programación, *front-end* (del lado del cliente) y *back-end* (del lado del servidor). Las instrucciones del lado del cliente consisten en códigos *JavaScript* que se ejecuta en el navegador. Por otra parte, las instrucciones que se ejecutan del lado del servidor son instrucciones escritas en lenguajes de scripting o programación, como *ASP.Net, PHP, Python*, por mencionar algunos, que son ejecutadas para crear lo que el usuario ha solicitado en su interacción con la página.

2.1.4. Bancos de datos Web

El término banco de datos Web es un concepto que se ha popularizado en el periodo 2010-2020. La explosión y aceptación tecnológica de herramientas informáticas en la industria ha permitido que este concepto gane cada vez más terreno (Moreno, 2014). Los datos generados por una variedad de dispositivos conectados a internet (computadoras, sensores, teléfonos inteligentes, electrodomésticos, equipos biométricos e industriales) permite que se genere datos exponencialmente, creando estos bancos de datos Web. Estos datos pueden ser almacenados y publicados en la Web para consumo de investigadores.

La información contenida en los bancos de datos Web se clasifica en dos tipos: a) datos estructurados los cuales están organizados y formateados en forma consistente, lo cual permite representarlos fácilmente en bases de datos relacionales en los que cada elemento es representado siempre con los mismos atributos. b) datos no estructurados los cuales no tienen una organización definida, es decir cada entidad representada en la base de datos puede ser descrita por diferentes atributos o incluso por texto libre, lo cual hace complejo su manejo (Sagiroglu & Sinanc, 2013). Los bancos de información no estructurada se han popularizado, ya que cuentan con la flexibilidad para poder ser utilizados por aplicaciones, sistemas y metodologías de diferentes rubros, sin embargo, el uso desmedido de estos datos puede provocar serios problemas de análisis cuando se ocupan de forma irresponsable (Bessis & Dobre, 2014).

La información que se maneja en los bancos de datos Web es de gran valor. Las enormes cantidades de información que se producen brindan la oportunidad de encontrar datos que sirvan para mejorar actividades relacionadas con el tipo de información contenida en estos bancos. Cabe mencionar que los entornos de datos Web se caracterizan por las 4 Vs que también caracterizan los entornos Big Data: Volumen de información, Velocidad de crecimiento, Variedad de orígenes y formatos, y el Valor de la información. Estas características hacen imposible la realización de todas las tareas de búsqueda, descarga y procesamiento de forma manual en tiempos concretos (Sagiroglu & Sinanc, 2013).

2.2. Web Scraping

El *Web Scraping* (Raspado Web) es una técnica de minería de datos por la cual se obtiene datos de páginas Web de forma automática. Estas técnicas arañan o raspan los diferentes sitios Web en busca de información por medio de bloques de código o robots llamados arañas. Estos mecanismos intentar replicar la actividad que realiza un usuario cuando información en una página Web (Bo, 2017). El *Web Scraping* se centra en tres aspectos importantes, la búsqueda de la información, la extracción de datos y el almacenamiento de datos para un análisis posterior (Vargiu & Urru, 2013).

El *Web Scraping* utiliza diferentes herramientas para generar un producto de valor: Protocolo de transferencia de hipertexto (por sus siglas en inglés (HTTP)), Lenguaje de Marcado Extensible (por sus siglas en inglés (XML)), valores separados por comas (por sus siglas en inglés (CVS)) y notación de objeto de *JavaScript* (por sus siglas en inglés (JSON

El mercado de las nuevas tecnologías ha permitido la creación de múltiples librerías y frameworks que nos facilitan la aplicación de métodos abstractos por medio de *APIs*. Las bibliotecas como *HtmlSQL, Requests, Guzzle, Goutte* facilitan la integración con bases de datos permitiendo utilizar protocolos HTML combinado con solicitudes *POST, GET* y utilizando formatos XML y JSON para la trasferencia de archivos. Otras herramientas como Scrapy (Scrapy.org, s.f.), *BeautifulSoup* (Richardson, s.f.) , Import.io (import.io, s.f.) permiten la integración de componentes más elaborados y que son necesarios para desarrollar aplicaciones de *Web Scraping* que sean eficientes.

2.2.1. Técnicas de *Web Scraping*

Los sistemas de *Web Scraping* (raspador Web) se componen de tres partes fundamentales (Bo, 2017):

- Rastreador (Crawler por su traducción al inglés): Recorren los enlaces en la Web usando un sitio de partida y permite crear copias del contenido de los sitios visitados, de manera similar a un motor de búsqueda.
- Raspador (Scraper por su traducción al inglés): Realizan la extracción de información de sitios específicos, buscando expresiones regulares, palabras clave, elementos, atributos, entre otros.
- Araña Web (Spider por su traducción al inglés): las arañas Web permiten iterar a través de los enlaces en las páginas Web hasta el nivel de profundidad indicado. Los

enlaces son identificados mediante sus etiquetas por lo que es requerido un análisis sintáctico del HTML.

2.2.1.1. Extracción manual

La extracción de información de forma manual de un sitio Web se centra en la revisión, selección y descarga de información sin la ayuda de un programa informático. Las principales actividades son las de copiar y pegar información del sitio o en otros casos brindar información en formularios e interactuar con la página hasta localizar los elementos que se desean recuperar (Mendoza , 2011).

Esta técnica de recolección de datos causa múltiples errores en la descarga, ya que no se lleva un control claro de lo que se está recuperando, por lo cual el usuario puede descargar un mismo dato múltiples veces. Esto se convierte en una actividad laboriosa que termina arruinando los proyectos donde se ocupa la información recuperada.

2.2.1.2. Análisis sintáctico HTML

La revisión y análisis del lenguaje de hipertexto permite visualizar la composición del sitio Web del cual se pretende sacar la información. Existen herramientas, como XQuery, HTQL, BeautifulSoup (Richardson, s.f.), que permiten la utilización de funciones preprogramadas que realizan la separación y clasificación de los diferentes elementos contenidos transformando el código en texto manipulable (Mendoza , 2011).

Las opciones para la manipulación de lenguaje HTML son variadas. Las hojas de contenido de estilo permiten seleccionar los elementos HTML que se requieren de acuerdo con las propiedades de clase que contengan estos elementos. XPath es otra opción similar al lenguaje CSS que permite seleccionar múltiples selectores. Los patrones URI permiten seleccionar recursos añadiendo expresiones regulares a la selección de elementos, la principal diferencia con los dos anteriores es que en URI se pueden seleccionar documentos y no sólo un elemento único. Visual Selector son otra opción de manipulación HTML la cual se centra en elegir los nodos HTML que contengan elementos visuales y las propiedades dadas por el navegador, identificando los elementos que se conecten con el mismo nodo y obteniendo la información contenida en elementos de la misma clase.

2.2.1.3. Expresiones regulares

Las expresiones regulares son cadenas de texto con símbolos diferentes que son utilizados para localizar patrones dentro de una cadena de texto. Los patrones de texto son evaluados por la expresión generada identificando las partes que coincidan dentro de un texto o documento especifico (Microsoft, s.f.).

Las expresiones regulares son utilizadas cuando se requiere encontrar un dato específico dentro de documentos extensos. Las herramientas disponibles para aplicar este tipo de técnica son escasas, pero las existentes ofrecen una gama de clases que nos permiten realizar una búsqueda y manipulación de datos realizando la comparación de expresiones regulares. RegExp es una de las herramientas más utilizadas por su amplia documentación y facilidad de agregación a nuevos proyectos.

2.2.1.4. Análisis del Modelo de Objetos del Documento

El análisis de un *Document Object Model* (por sus siglas en ingles DOM) incorporado a un navegador Web permite la recuperación de contenido dinámico generado por scripts. Los controles del navegador analizan el árbol DOM para obtener partes del sitio Web de interés. La secuencia de comandos generada del lado del cliente permite analizar la página Web y recuperar partes de esta (Uzun, 2020). Los reconocimientos de anotaciones semánticas son una opción especial del análisis de árboles de objetos. La aplicación de este método requiere que las páginas Web contengan metadatos, anotaciones y marcas semánticas que son utilizadas para la localización de fragmentos de datos específicos.

2.2.1.5. Agregación vertical

Las plataformas Web que tienen características que permiten la creación de numerosos programas (bloques de código) enfocados a mercados específicos. Esta técnica realiza el establecimiento de bases de conocimiento de plataformas verticales, las técnicas de este tipo se caracterizan por su robustez y calidad de datos. Sin embargo, es necesario identificar cada elemento del sitio Web utilizando aprendizaje automático. Es decir, la aplicación de esta técnica requiere de la existencia de marcas y anotaciones que sirvan como punto de referencia para el contenido, la manera más común es utilizar los marcadores CSS como punto de referencia (Hernández, y otros, 2015).

La Tabla 1 muestra una comparativa de las técnicas de *Web Scraping* que se pueden utilizar para la extracción de información.

Tabla 1.Técnicas de Web Scraping

Técnica de raspado	Descripción	Tipo de escaneo	Herramientas
Extracción manual	El usuario navega por los sitios buscando la información de interés	Manual	No
Expresiones regulares	En esta técnica se programa una serie de expresiones de caracteres y se comparan con el contenido de los sitios Web	Automático	RegEx, regexp
Programación del protocolo HTTP	Las solicitudes del protocolo se implementan por medio de programación para obtener bloques de datos	Automático	Requests, urllib
Análisis del lenguaje de marcado de hipertexto HTML	Esta técnica permite ver la composición del sitio Web del cual se pretende sacar la información, permitiendo clasificar y separar los contenidos	Automático	BeautifulSoup, Xquery, HTQL, Scrapy, Selenium
Análisis de documentos DOM	Esta técnica permite la recuperación de contenido generado por scripts. En estas técnicas se construyen documentos de tipo árbol por los cuales se puede navegar hasta un elemento en concreto.	Semiautomático	ECMAscript,CORBA,JAVA IDL
Analizadores con visión por computadora o agregación vertical	Los analizadores utilizan aprendizaje automático para identificar cada elemento del sitio Web.	Semiautomático	ScraperWiki, PHP, Guzzle,Jsoup
Escaneo de micro formatos	Son porciones de código cuyo objetivo es insertar recursos dentro del sitio Web los cuales pueden insertar contenido semántico que sirven como identificadores de clases específicas. Estas con utilizadas para identificar datos importantes.	semiautomático	Microdata, RDFa, Microformats

Programas comerciales	Son herramientas que ya implementan las diferentes técnicas de *Web scraping* en sus operaciones	Semiautomático /automático	No

2.3. Métricas de medición

A continuación, se presenta un conjunto de métricas que son utilizadas para evaluar y medir la efectividad de las técnicas de Web Scraping, en los experimentos de extracción de información de sitios Web.

2.3.1. Precisión

La métrica de precisión representa el número de resultados correctos obtenidos en una extracción de información. Es decir, se obtiene el número de archivos correctamente identificados de los archivos esperados por cada extracción realizada por el sistema en comparación con los archivos esperados para cada recuperación de datos (Parra, 2016). El valor de precisión se representa como se muestra en la Fórmula.1 que se muestra a continuación.

$$\text{Precisión} = \frac{AR}{AR+ANR}$$

Fórmula 1.Cálculo de precisión

Donde:

AR: Total de archivos recuperados

ANR: total de archivos no recuperados

2.3.2. Exhaustividad

La métrica de exhaustividad devuelve el porcentaje de los archivos con información correcta recuperados con respecto al total de archivos con información correcta existente en las bases de datos (Mendoza , 2011). La fórmula 2 describe la métrica.

$$\text{Exhaustividad} = \frac{AR}{TA}$$

Fórmula 2. Cálculo de Exhaustividad

Donde:

AR: Total de archivos recuperados

TA: Total de archivos esperados

2.3.3. Valor F1

La medida F1 obtiene el promedio entre los valores de precisión y exhaustividad. Esta medida permite comparar el rendimiento de estos dos valores con varias soluciones posibles. En la Formula 3 se describe este valor.

$$F1 = 2 * \frac{Precisión * exhaustividad}{Precisión + exhaustividad}$$

Formula 3 Calculo de valor F1

Las pruebas de recolección de datos utilizando técnicas Web Scraping utilizan el valor F1 como medida de éxito o fracaso en la extracción de datos. Ya que esta nos permite ver de forma cuantitativa cuantos elementos se han logrado obtener en las pruebas de rendimiento.

Capítulo 3

Estado del Arte

A continuación, se describe una serie de trabajos que están estrechamente relacionados con el proyecto de tesis. Con la finalidad de homogeneizar la descripción de cada uno de los trabajos del estado del arte, se han utilizado los siguientes aspectos:

- **Descripción:** se mencionan las características del trabajo de investigación, los puntos claves de la investigación y los experimentos realizados.
- **Conclusiones:** se mencionan las aportaciones de cada trabajo, haciendo énfasis en las partes de ese trabajo de estado del arte que pueden aportar valor a proyecto de tesis.

3.1. "Raspado Web"

3.1.1. Descripción

El trabajo presentado por Zhao (Bo, 2017) se describe las características y componentes de un sistema de Web *scraping* (Raspado Web). Zhao describe *Web Scraping* como una técnica para extraer datos de internet y colocarlos en una base de datos para un análisis posterior. De acuerdo con el autor, los raspadores Web tienen la capacidad de adaptarse a diferentes escenarios, organizando la información extraída de forma automática.

Los experimentos realizados por los autores generaron solicitudes HTTP para crear la conexión con el sitio Web que se utilizó como caso de estudio. En este trabajo se utilizaron bibliotecas que permiten la implementación de servicios Web y conexiones con las fuentes de información, como lo son Urllib y Selenium. De esta forma, se logró recuperar datos que pueden ser visualizados por los usuarios para la selección de datos de mayor valor. La selección de los datos adecuados se realizó con bibliotecas con procesos de extracción repetitiva como BeautifulSoup y scrapy.

El autor hace recomendaciones en cuanto a las restricciones legales del manejo de información. la advertencia que plasmas es para prevenir y alertar a los desarrolladores de las consecuencias del uso desmedido de las técnicas Web Scraping. Los derechos de autor, condiciones de servicio, saturación de solicitudes y el bloqueo de acceso total de la IP del ordenador en el que se realiza la extracción son las principales sanciones de los Web scraping cuando no se implementan de manera regulada.

3.1.2. Conclusiones

Esta investigación pone las bases de la implementación de un sistema de *Web Scraping* (Raspado Web). El artículo describe las diferentes funciones y tareas de esta tecnología, tomando en cuenta la creación de diferentes módulos para la búsqueda, extracción y almacenamiento de datos.

Los autores de este trabajo describen la estructura básica de un sistema de Web Scraping, listando diferentes herramientas que son de utilidad para su implementación. El artículo ofrece recomendaciones de buenas prácticas en cuanto a extracción de datos para prevenir los bloqueos de acceso de los diferentes sitios y define técnicas para incurrir en problemas legales

3.2. "Investigación y desarrollo de técnicas de raspado"

3.2.1. Descripción

El trabajo realizado por Rodríguez (Villanueva Rodriguez, 2019) presenta un análisis de herramientas basadas en técnicas de *Web Scraping* (Raspado Web) que estaban activas en el año 2019. El artículo muestra un conjunto de herramientas que se utilizaron para el análisis de 18 sistemas, los cuales fueron comparados según los servicios que ofrecían, la personalización de las búsquedas, la interfaz gráfica, los complementos de inteligencia artificial, el plan de pagos, la documentación disponible, el tamaño de la comunidad de usuarios y los repositorios públicos.

El artículo presenta pruebas de búsqueda de datos en documentos utilizando la librería Jsoup para la implementación de técnicas de *Web Scraping*. Los enfoques se pusieron a prueba analizando un fragmento de una página Web. Se obtuvieron resultados favorables utilizando los métodos DOM, la selección por sintaxis, la manipulación URL y los patrones estratégicos. Los resultados de las diferentes pruebas permitieron obtener 100 mil resultados aplicando un enfoque recursivo, depurando los datos con servicios Web y aplicando a diferentes patrones para e-commerce.

3.2.2. Conclusiones

Este trabajo de investigación presenta una descripción de cada una de las herramientas utilizadas para la creación de sistema de *Web Scraping*. La clasificación que realiza el trabajo muestra un acercamiento a las ventajas y desventajas de cada sistema como lo es documentación publicada, capacidad de extracción, métodos de licencias, tipos de extradición, de esta forma, el lector pueda comparar las características de cada sistema y poder elegir uno de acuerdo con sus necesidades.

La prueba de los enfoques mostró la implementación de los procesos de un *Web Scraping* (Raspado Web). Cada una de las pruebas realizados por los autores del trabajo generaron resultados positivos, logrando definir un método de tres pasos. El primer paso es la obtención de URLs personalizando el sistema, el segundo paso es la búsqueda y filtrado de los resultados según el contenido que se esté buscando. Finalmente, el tercer paso consiste en el procesamiento de los datos y la exportación a formatos estandarizados

3.3. “Un enfoque novedoso de Raspado Web usando información adicional obtenida de páginas Web”

3.3.1. Descripción

Los trabajos realizados por Uzun (Uzun, 2020) proponen un nuevo enfoque, denominado UzunExt, para implementar técnicas de *Web Scraping* (Raspado Web). UzunExt es un enfoque que se centra en acelerar los procesos de extracción de información realizando los procesos en periodos de tiempo corto y utilizando menos recursos. El enfoque propuesto consta de dos componentes; el primero rastrea las páginas Web y el segundo: extrae los datos encontrados. Adicionalmente, se recolecta información durante el rastreo de la página que es utilizada, estos datos son utilizados para aumentar la eficiencia con respecto a los tiempos de extracción.

El enfoque UzunExt pretende dejar de lado la creación de un árbol de dominio (DOM por sus siglas en inglés). Los métodos de cadena son la solución implementada por Uzun (Uzun, 2020), este método tiene seis parámetros de entrada; etiqueta de apertura, fuente, posición inicial, numero de etiquetas anidadas, repetición del nombre de la etiqueta y un sistema de búsqueda.

UzunExt se dividen en tres sistemas principales: el primer sistema inicializa los procesos de extracción dando un repaso a la composición del sitio de estudio. El segundo sistema implementa la extracción utilizando los métodos de UzunExt. El último sistema es el encargado de encontrar el valor apropiado para la posición inicial de la extracción. La predicción del punto adecuado para la extracción se puede determinar examinando al menos dos páginas Web de un mismo sitio Web.

3.3.2. Conclusiones

Los autores realizaron experimentos de funcionalidad poniendo a prueba su enfoque con diferentes sitios Web. Los 100 sitios Web utilizados contenían datos en diferentes idiomas y formatos. Los resultados fueron favorables al utilizar el enfoque UzunExt obteniendo 300 páginas y 4 patrones de las búsquedas realizadas, mejorando 2.35 veces la velocidad de otros métodos.

3.4. "Raspado Web basado en la nube para aplicaciones de *Big Data*"

3.4.1. Descripción

El trabajo realizado por Ram Sharan, Chaulagain y compañía (Ram Sharan, Santosh, & Sadhu Ram, 2017) analiza el proceso de extracción de información de páginas Web con contenido no estructurados. Algunos de los problemas que se pueden encontrar en este tipo de contenido son las restricciones que tienen los sitios web como son el captcha, métodos de validación de usuarios, almacenamiento de volúmenes de datos muy grandes, datos extraviados. Si embargo el autor presenta la utilización de herramientas y servicios en la nube, como los servicios Web de Amazon elastic compute cloud y DynamoDB. La herramienta elegida por la autora para la implementación de extracciones de información fue Selenium el cual es un programa que permite trabajar con páginas web eh implementar pruebas de funcionalidad.

3.4.2. Conclusión

La solución propuesta presenta la utilización de servicios de Amazon en conjunto con la biblioteca Selenium creando de esta forma pruebas de extracción con navegación real simulada. Los servicios permiten restringir los recursos y tener mejor control de la información.

3.5. "Tecnologías de Raspado Web en un mundo API"

3.5.1. Descripción

El trabajo realizado por Daniel Glez-Peña (Glez-Peña, Lourenc¸o, López Fernández, Reboiro Jato, & FdezRiverola, 2013) hace una revisión de herramientas que aplican el *Web Scraping* (Raspado Web) identificando fortalezas y limitaciones al realizar la tarea de extracción de información. La técnica de *Web Scraping* es descrita como una alternativa automática para la obtención de información, la cual imita la interacción entre los servidores Web y los seres humanos. Las API y los frameworks realizan las tareas más comunes para la obtención de datos Web. Las tareas que mencionan se limitan a tres puntos clave:

El primer paso se refiere al acceso al sitio Web utilizando el protocolo de transferencia de hipertexto (HTTP por sus siglas en inglés) y aplicando métodos GET y POST para la obtención y envió de la información de la Web. El segundo paso presenta el análisis HTML y la extracción de la información contenida utilizando librerías y selectores de contenido

como XPath y CSS. El tercer paso se refiere a la presentación de resultados, donde se busca la transformación de los datos presentándolos de forma estructurada para permitir su análisis posterior.

El artículo presenta las librerías y frameworks utilizados para el desarrollo de sistemas de *Web Scraping,* presentando el tipo de herramienta, el dominio del lenguaje, identifica si se trata de una API o es una herramienta independiente, el lenguaje en el que está implementada, y el tipo de extracción que realiza. Las herramientas tomadas en cuenta son; *UNIX sheell, Curl o libcurl, Web -harvest, Jsoup, HttpClient, jARVEST, WWW:Mechanize, Scrapy, BeautifulSoup*, las cuales fueron comparadas de acuerdo a las características mencionadas anteriormente..

3.5.2. Conclusiones

En este trabajo de investigación se utilizaron servidores como WhichGenes y PathJam aplicando las técnicas para obtener datos de estos sitios. WhichGenes y PathJam son servidores que contienen información de bioinformática. WhichGenes recupera listas de genes relacionados con enfermedades metabólicas. *PahtJam* es un metaservidor dedicado a la integración de bases de datos de vías genéticas.

Como resultado de la experimentación, se obtuvieron 315 resultados de 7 sitios Web. Los resultados se presentaron en una tabla mostrando el tipo de proceso que siguió el Web Scraping (Raspado Web) para obtener la información. Se concluye que la aplicación de robots de extracción para datos muy específicos tiene mejor desempeño al implementar robots de extracción dedicados.

3.6. “Una descripción general de las técnicas y herramientas de Raspado Web”

3.6.1. Descripción

El trabajo presentado por Anand V. Saurkar (Saurkar, Pathare, & Gode, 2018) describe diversas técnicas de Web Scraping (Raspado Web) y menciona algunas herramientas que ayudan a la implementación de estas técnicas. Los aspectos importantes de ese concepto son descritos y comparados entre sí.

Los enfoques que se presentan son solicitudes HTTP, manipulación HTML, creación de árboles de dominio DOM, análisis de pantalla, entre otras. Las herramientas presentadas para facilitar los enfoques mencionados son: Mozenda, Web Content Extractor, Import.io las cuales ofrecen una serie de herramientas para la implementación rápida de las tareas de un sistema de este tipo.

Las áreas de aplicación de este tipo de sistemas son detección de productos, comparación de precios, monitoreo de información, análisis de datos, análisis de mercado, entre otras. Los métodos de Web Scraping (Raspado Web) facilitan la extracción de los sitios que manejan las áreas mencionadas, disminuyendo los tiempos y maximizando los recursos.

3.6.2. Conclusiones

Este artículo de investigación ofrece una introducción al concepto de Web Scraping (Raspado Web) como técnica para la extracción de datos. La mención de los enfoques y herramientas de desarrollo permite al lector obtener las características básicas para implementar un sistema basado en estas técnicas.

3.7. “Herramientas para recopilar la información en las noticias publicadas en los sitios Web de internet”

3.7.1. Descripción

El trabajo propuesto por Ariannis Vargas Pérez (Vargas Pérez, 2019) propone el desarrollo de una herramienta informática que permita llevar a cabo el proceso de monitoreo y seguimiento de información contenida en publicaciones Web y sus comentarios. La herramienta dota al usuario de información de páginas Web, actualizando la base de datos con información reciente y el envío de notificaciones por correo electrónico. Cabe mencionar que se realiza una revisión de la técnica de Web Scraping (Raspado Web) resaltando algunas características y ventajas de la tecnología.

La herramienta propuesta por los autores busca la automatización de los procesos realizados por el centro Ideo-informática (CIDI). La recopilación y actualización de información son las actividades principales de este organismo, la solución que se plantea es la utilización de métodos de Web Scraping (Raspado Web) combinando tres tareas básicas los Crawler o Rastreador, Spider o araña y el scraper o extractor.

3.7.2. Conclusión

La herramienta generada en este trabajo de investigación cuenta con tres funcionalidades principales. La primera consiste en un módulo de recopilación, el cual se encarga de identificar las publicaciones realizadas guardando datos de referencia como son; titulo, dirección, autor, cantidad de visitas, fecha, fuente de orígenes y comentarios. El segundo módulo es de actualización, el cual se encarga de renovar la información recopilada actualizando la cantidad de visitas, la cantidad de comentarios y cambios en general del cuerpo de la publicación. Por último, el tercer módulo es de generación de reportes, que muestra un resumen de la información recopilada.

Los experimentos realizados por los autores de este trabajo arrojaron resultados positivos para diferentes casos. El método manual recopiló información con un promedio de 50 a 90 segundos por cada extracción. La obtención de los comentarios requirió un promedio de 25 a 40 segundo por extracción. El artículo presenta una muestra de 7 publicaciones y 144 comentarios en 1,7 horas para la extracción manual de la información. La extracción de la misma muestra utilizando la herramienta mejoro a 11,8 minutos para la obtención total del contenido, lo cual implica un ahorro significado en el tiempo de recuperación de información

3.8. “Web Scraping simplificado con SiteScraper”

3.8.1. Descripción

El trabajo propuesto por Richard barón y compañía (Penman, Baldwin, & Martínez, 2010). Propone un *Web scraper* que recupere patrones basados en Xpath que permitan identificar listas definidas por el usuario dentro de una estructura HTML. Los autores ingresan un conjunto de páginas Web que sirven como datos de entrenamiento para la detección de patrones que parten de cadenas de datos que el usuario busca recuperar. De esta forma el *scraper* genera diferentes consultas Xpath que describen donde encontrar la información que se busca.

3.8.2. Conclusión

La herramienta que presentan en el artículo realiza tres procesos para la generación de un patrón Xpath. El primero es analizar gramaticalmente el documento HTML de la página que se está utilizando, con el fin de buscar la semilla que sirva como punto de partida para la

creación del patrón de búsqueda. El scraper divide el documento HTML en nodos individuales y obtiene los contenidos asociados a cada uno de ello. De esta forma se obtiene una posición relativa del tipo de elemento que se encuentra en ese nodo. El segundo proceso es identificar el elemento que contiene cada fragmento de texto asociado a la semilla inicial de esta forma se genera una tabla con la cual se va indexando cada elemento con la cadena generada, eliminando etiquetas que sólo son para el formato como saltos de línea, espacios en blanco, tabulaciones, entre otras etiquetas. El tercer paso es generalizar las ubicaciones que mejor coinciden y combinarlas cuando sea necesario, de esta forma el scraper identifica cuando el usuario quiere todos los elementos disponibles o sólo una parte de ellos identificando brechas entre los elementos seleccionados. SiteScraper tenía dos objetivos originales, raspar de forma fácil datos detallados y la posibilidad de adiestramiento del raspado con cambios automáticos en la estructura para nuevas páginas Web

3.9. Conclusión

Los sistemas basados en técnicas de Web Scraping (Raspado Web) son variados. Sin embargo, cada uno de los sistemas y herramientas buscan facilitar las mismas tareas de localización, recolección y análisis de la información. Las características que más se toman en cuenta son la velocidad de extracción, la capacidad de manipulación de datos y el preanálisis de los datos.

La implementación correcta de estas características asegura la creación de un sistema de calidad. El cual obtendrá los mejores resultados según sea la prioridad del usuario que implemente las técnicas de Web Scraping (Raspado Web), mayor velocidad, mayor número de resultados o en su caso un pre análisis adecuado. La decisión de usar un enfoque u otro y la utilización de una herramienta queda a decisión del usuario según sean sus prioridades, la Tabla 2 nos muestra las características principales de cada estudio consultado.

Tabla 2.Comparación del estado del arte

#	Título	Idea principal	Tecnologías usadas	Resultados	Aportación para el proyecto de tesis
1	Raspado Web (Bo, 2017)	Se presenta una descripción de la tecnología de Web Scraping (Raspado Web), abarcando las conexiones con páginas Web.	Urllib2, Selenium, BeautifulSoup, Pyquery, Scrapy, Nuthch, Import.io	Se muestra la realización de conexiones por medio de solicitudes HTTP para la recopilación de información	La introducción a la tecnología de raspado es presentada de forma sencilla, presentando características y procedimientos de aplicación.
2	Investigación y desarrollo de técnicas de raspado. (Villanueva Rodriguez, 2019)	La investigación muestra una comparación de diferentes herramientas que utilizan técnicas de Web Scraping (Raspado Web). Aplicando parámetros como servicios ofrecidos, plan de pagos, entre otros.	Servicios Web, Protocolos de transferencia de archivos, Mozenda, Scrapy, entre otros.	Los ejercicios de búsqueda se realizaron con una muestra de 100 páginas Web donde se comparó los resultados obtenidos en cada tarea para medir el desempeño de las herramientas comparadas.	El catálogo de herramientas permite visualizar las características de cada una de ellas, pre seleccionando aquellas que puedan ser de utilidad en la creación del sistema propuesto.
3	Un enfoque novedoso de Raspado Web usando la información adicional obtenida de páginas Web. (Uzun, 2020)	El autor propone un nuevo enfoque de búsqueda y extracción de datos llamado UzunExt. El cual tiene como base los métodos en cadena ara recopilación de información	No se menciona la tecnología utilizada	La muestra de pruebas fue de 100 sitios Web de características diferentes de las cuales se obtuvieron 300 páginas de interés y se generaron 4 patrones de navegación.	El documento muestra un posible enfoque para aplicar en el sistema propuesto.

4	Raspado Web basado en la nube para aplicaciones de macro datos. (Ram Sharan, Santosh, & Sadhu Ram, 2017)	Los enfoques de Web Scraping (Raspado Web) locales y basadas en la nube son comparados. Las características tomadas para el análisis son solución de problemas en extracción, captcha, manejo de volúmenes, fiabilidad de los datos.	Servicios Web de Amazon: elastic compute cloud y DynamoDB, Selenio, Scrape-hub	Un experimento realizado con una muestra de 50 sitios Web pertenecientes a Amazon, Google shopping, entre otros, los sistemas basados en la nube muestran mejores resultados en cuanto a rentabilidad, optimización de recursos, reducción de costos.	el diseño del sistema toma mayor valor enfocándolo como una aplicación Web, basado en tecnología en la nube las cuales son la tendencia de la actualidad
5	Tecnologías de Raspado Web en un mundo API. (Glez-Peña, Lourenc¸o, López Fernández, Reboiro Jato, & FdezRiverola, 2013)	Las fortalezas y limitaciones de herramientas de Web Scraping (Raspado Web) son explicadas y analizadas. Los problemas que pueden surgir son mencionados; fallas de conexión, denegaciones de servicio, bloqueos de IP, la demanda masiva de datos.	SOAP, REST, HTTP, solicitudes GET y Post, libcurl, Apache HttpClient, Jsoup, htmlcleaner, BeautifulSoup, scrapy, Web - Harvet	Las pruebas de rendimiento fueron realizadas con meta servidores de WhichGenes y PathJam las cuales cuentan con información de bio-información	Este artículo representó un buen punto de partida para la recolección de datos de entorno médico
6	Una descripción general de las técnicas y herramientas de Raspado Web. (Saurkar, Pathare, & Gode, 2018)	El proyecto muestra las características y usos de las técnicas del Web Scraping (Raspado Web). La lista de técnicas es mostrada con ejemplos para una mayor comprensión	Mozenda, Web content extractor, import.io	Los experimentos son mostrados utilizando las diferentes técnicas y los resultados obtenidos para cada caso.	El artículo presenta una lista de técnicas disponibles para la implementación de sistemas de Web Scraping: HTTP, analizadores HTML, HTQL, arboles de dominio DOM, análisis de pantalla

7	Herramienta para recopilar la información en las noticias publicadas en los sitios Web de internet. (Saurkar, Pathare, & Gode, 2018)	El trabajo de investigación propone una alternativa automatizada para la recolección y actualización de los datos de las paginas consultadas en la universidad de UCI.	Python, Web scraping. search, Web scraping. remove, *PostgresSQL, Aptana Studio, PyQt, Psycopg2*	Se obtuvo una herramienta dividida en tres módulos; recopilación, actualización y reportes los cuales permiten la recolección de información de noticias publicada en sitios Web reduciendo los tiempos y el esfuerzo necesario	El proyecto ofrece una posible estructura a aplicar en el proyecto de investigación permitiendo tener un punto de partida y una visualización de los resultados posibles con la implementación del sistema.
8	*Web Scraping* simplificado con SiteScraper. (Penman, Baldwin, & Martínez, 2010)	El trabajo propone la extracción de información generando patrones de diseño que permitan identificar el contenido desde nivel código.	BeautifulSoup	El programa logro crear diferentes patrones que sirven para la localización y descarga de información con los datos proporcionados.	Los patrones de identificación propuestos en este artículo fueron una base para el desarrollo del sistema desarrollado en esta tesis.

Capítulo 4

Metodología de Solución

4.1. Descripción general de la solución

Este trabajo de investigación muestra la implementación de un sistema informático basado en técnicas de *Web Scraping* que permiten la obtención de información de páginas Web de forma automatizada. El sistema realiza la conexión con el servidor Web donde se encuentra alojado el sitio Web del que se requiere recuperar información. Esto permite recuperar la estructura HTML de la página Web con la cual el sistema interactúa para extraer su contenido.

El sistema se centra en la recuperación de datos estructurados que se encuentran almacenados dentro de un sitio Web de datos abiertos, los cuales son puestos a disposición como recursos descargables y accesibles para todo público. La recuperación de esta información permite la creación de un conjunto de datos de un tema específico unificando la información de diferentes fuentes en una sola base de datos. La información descargada es almacenada y puesta a disposición del usuario del sistema para su aplicación o uso en diferentes procesos o aplicaciones.

La actualización de la información en la Web es constante, por lo cual esto requiere de descargas múltiples de los datos de las páginas Web a las cuales se les aplica *Web Scraping*. Por ello, el sistema realiza un monitoreo de la información publicada en los sitios Web donde se realiza el *Web Scraping*, detectando los cambios en la página Web y recuperando los nuevos datos disponibles. Esto permite actualizar el conjunto de información generado en la descarga múltiple de datos.

4.2. Sistema BDScraping

El sistema desarrollado en esta tesis ha sido denominado BDScraping. El sistema cuenta con cinco funcionalidades relevantes: configuración de búsqueda, búsqueda de información, recolección de información, descarga de información y monitoreo de información. Estas funcionalidades se desarrollaron en el lenguaje Python el cual cuenta con bibliotecas especializadas en técnicas de rastreo de información para extraer datos de páginas Web de forma automatizada. A continuación, se describen las funcionalidades del sistema BDScraping.

- Configuración de búsqueda. Con esta funcionalidad, el sistema recibe las características iniciales de una búsqueda de información las cuales se usarán para realizar los procesos de localización, extracción de información Web.
- Búsqueda de información. Con esta funcionalidad, el sistema inicia la conexión con la página Web. El sistema recibe las características iniciales de la búsqueda y valida la existencia del sitio Web ingresado. De esta forma se valida que los sitios Web proporcionados por el usuario existen dentro de la Web.

- Recolección de información. Con esta funcionalidad, el sistema analiza la página Web en busca de los elementos a descargar. En esta funcionalidad se utilizan las rutas XPath configuradas por el usuario para la ubicación de los elementos que se requiere descargar.
- Descargas de información. El sistema realiza las descargas de la información recolectada del sitio Web y comienza a almacenarla en el espacio especificado.
- Monitoreo de la información. en esta funcionalidad el sistema realiza revisiones constantes a los sitios Web de los cuales se obtuvo información. Esto se realiza para buscar cambios en los datos recolectados, actualizaciones en las estructuras o la eliminación de la información almacenada en el sitio Web.

Las funcionalidades se concentran en la manipulación de la estructura HTML del sitio Web, realizando interacciones con la página Web utilizando los requisitos de configuración para la recuperación de información. Estas interacciones permiten la extracción de información de diferentes sitios Web, con la cual se crea una base de datos única con información de un mismo tema, colocando la información recuperada bajo una misma estructura de almacenamiento. En la Figura 1 se presenta las funcionalidades del sistema BDScraping.

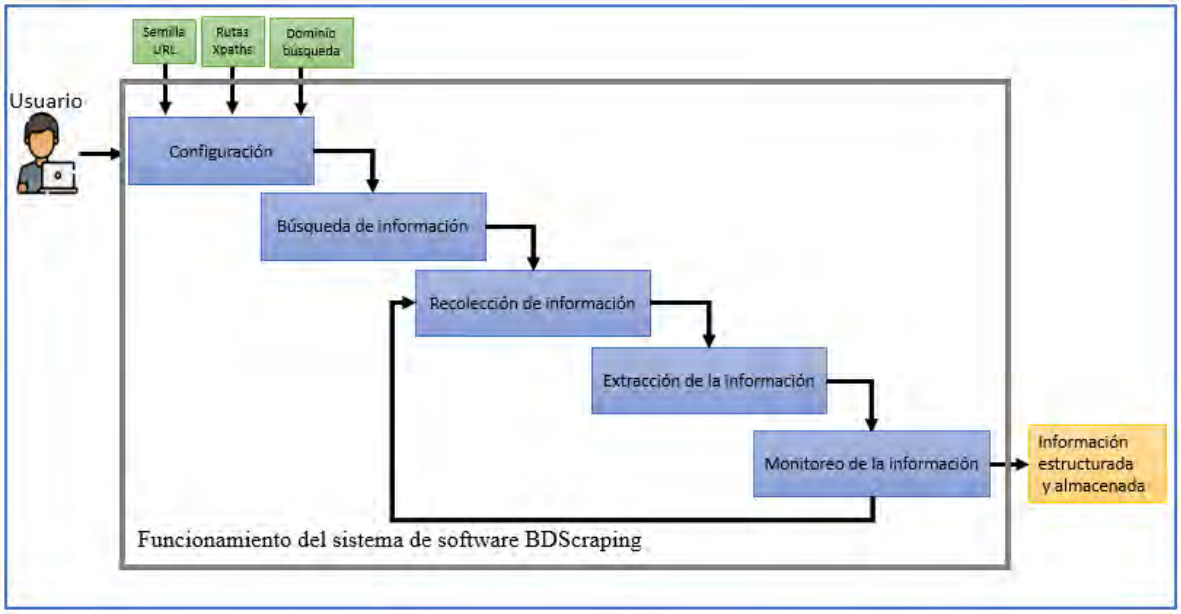

Figura 1. Modelo SADT del funcionamiento del sistema BDScraping

4.2.1. Configuración de búsqueda

La primera funcionalidad del sistema *BDScraping* (Figura 1) tiene como objetivo la configuración de las búsquedas de información que se realiza por el usuario del *BDScraping*. La configuración de la búsqueda requiere de cinco características iniciales las cuales se realizan de la siguiente manera:

- **Determinación del nombre de la búsqueda**, este nombre permite diferenciar los proyectos de búsqueda que el usuario configure dentro del *BDScraping*, de esta forma se pueden configurar múltiples búsquedas de un mismo domino sin afectar las características individuales de cada una de ellas.
- **Especificación de la base de datos** donde se almacenará la información que recupere el *BDScraping*. el utiliza el nombre de la base de datos para realizar una conexión y guardar la información recuperada, en caso de que la base de datos no exista el sistema crea una nueva base de datos con la información ingresada. En caso de que diferentes proyectos tengan una misma base de datos ambos proyectos ingresaran la información encontrada dentro de la misma.
- **Especificación de la URL de la página Web** que se utiliza para recabar la información, esta dirección permite al sistema realizar solicitudes de acceso y entablar una conexión con la cual se pueda obtener la información que busca el usuario.
- **Especificación del dominio de búsqueda** que se utiliza para determinar en qué páginas Web se aplicaran las configuraciones de la búsqueda, esta característica se encuentra dentro de la URL del sitio Web el cual agrupa a las páginas Web que tienen una misma configuración en su estructura, esto permite que la información que se encuentra publicada pueda manejarse de la misma forma y no obtener errores de búsqueda.
- **Especificación de la ruta Xpath** de los elementos que se quieren recuperar. Estas rutas son estructuras de sintaxis compuestas de nodos o etiquetas HTML con las cuales es posible navegar dentro de un sitio Web a nivel código, de esta forma el sistema puede ubicar elementos específicos dentro del árbol de dominio DOM de la página que se esté utilizando.

La Tabla 3 presenta un ejemplo de los requisitos para la configuración de una nueva búsqueda.

Tabla 3. Ejemplo de configuración de búsqueda

Nombre de búsqueda	Base de datos	URL de la página Web	Dominio de búsqueda	Rutas Xpath
Búsqueda	Base enfermedades	https://paginaWeb .com/enfermedades	PaginaWeb .com	//*[@id="ejemplo"] /a

Los bloques de código que se muestran en la Tabla 4 presentan la forma en la que se recuperan los valores iniciales de cada búsqueda de información con el *BDScraping*.

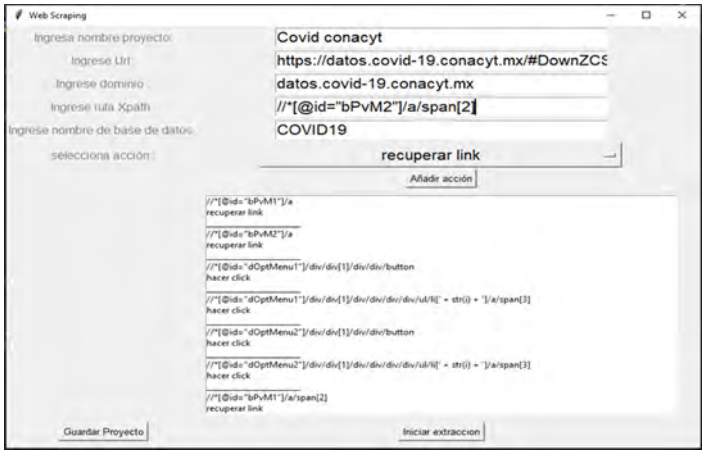

Figura 2 pantalla de configuración del BDScraping

El *BDScraping* genera variables de entrada, las cuales reciben los valores iniciales temporalmente hasta ser enviados a la base de datos de las configuraciones de búsqueda. Las primeras 5 variables pertenecen a cada valor inicial, la última variable es la que permite la interacción con la estructura la cual es asignada a las rutas XPath ingresadas.

Tabla 4. Ejemplo de codificación de inicialización de variables

```
url = tkinter.StringVar()

xpath = tkinter.StringVar()

nombre = StringVar()

base = StringVar()

dominio = StringVar()

vmenu =StringVar()
```

Los valores ingresados en el sistema son recuperados desde la interfaz gráfica realizando la unión entre la variable y el dato ingresado en las cajas de texto como se muestra en la Tabla 5.

Tabla 5.ejemplo de recuperación de datos ingresados

```
Label(miFrame,text="Ingresa nombre proyecto:", fg="gray",font=("Poppins",
12)).grid(row=0, column=0, pady=5)

urlE = Entry (miFrame, textvariable=nombre, font=("Poppins", 16),
width=35).grid(row=0, column=1, padx=5)
```

El ultimo valor es una opción que el *BDScraping* da al usuario para aplicar una de tres interacciones posibles para las rutas *Xpath* ingresadas de esta forma el *BDScraping* navega a través de la página Web. Las tres interacciones posibles un clic, recuperación de atributo,

Tabla 6.ejemplo de creación de botones y acciones

```
opt=OptionMenu(miFrame,vmenu, *listaOp)

opt.config(text="selecciona acción", font=("Poppins",16),width=35)

opt.grid(row=5, column=1, padx=5)

botonIngresar = Button (miFrame, text="Añadir acción", fg="black",
font=("Poppins", 10), command=enviarAccion)

botonIngresar.grid(row=6, column=1)
```

4.2.2. Búsqueda de información

La segunda funcionalidad del *BDScraping* (Figura 1) tiene como objetivo iniciar el proceso de conexión y validación con el sitio Web. el sistema utiliza los valores de URL y dominio para realizar una conexión con el sitio Web, de esta manera, el sistema recibe una respuesta del servidor donde se encuentra ubicada la página con lo cual se determina si existe, o si está habilitado o deshabilitado el sitio Web. La Tabla 7 se presenta un fragmento de código con el cual se recibe la URL del sitio Web y se prepara la inicialización de la página Web.

Tabla 7.Ejemplo de configuración de navegador Web

```
start_urls = [URL Semilla]

options = Web driver. ChromeOptions()

options.add_argument('--start-maximized')

options.add_argument('--disable-extensions')
```

El *BDScraping* divide la búsqueda de información en tres tareas iniciales las cuales se describen continuación.

1) La primera tarea, el *BDScraping* realiza requerimientos de acceso a los servidores donde se encuentra alojada la página Web. El requerimiento permite validar la existencia del sitio Web utilizado, si la URL proporcionada no existe, se envía una notificación al usuario de “URL no encontrada”. En caso contrario, el sistema extrae una copia del sitio Web (su contenido HTML) en un archivo con extensión .XML. La Tabla 8 presenta un fragmento de código de este proceso.

Tabla 8.Ejemplo de inicialización de navegador Web

```
def abrirnavegador()

    Web  = str(url.get())

    print(Web )

    if(Web ==True)

       driver = Web driver.Chrome("./chromedriver.exe", options=options)

       driver.get(Web )

       time.sleep(7)

else

       print ("URL no encontrada")
```

2) La segunda tarea, el *BDScraping* inicia un navegador Web de forma automática y en modo de prueba con lo cual elimina extensiones o complementos que se tengan instalados en el navegador Web mejorando el rendimiento y procesamiento de la información, mostrando al usuario la página Web con cada uno de los elementos que contiene la estructura HTML, de esta forma la página Web puede ser vista y manipulada por el sistema de *BDScraping*. La Tabla 9 presenta un fragmento de código de este proceso.

Tabla 9. Ejemplo de apertura de página Web

```
options = Web driver.ChromeOptions()

options.add_argument("--window-size= 300,300")

driver = Web driver.Chrome("./chromedriver.exe", options=options)

driver.get(Web )
```

3) La tercera tarea, el *BDScraping* realiza un mapa del contenido de la página Web en el cual se identifican cada uno de los elementos de la estructura HTML. Este mapa se utiliza como referencia de comparación para el monitoreo de la información. La Tabla 10

presenta un fragmento de código de cómo se realiza el mapa de la estructura HTML del sitio Web.

Tabla 10.Ejemplo de creación de mapa de un sitio Web

```
while True:

    try:

        response = urlopen(url).read()

        currentHash = hashlib.sha224(response).hexdigest()

        time.sleep(30)

        response = urlopen(url).read()

        newHash = hashlib.sha224(response).hexdigest()

        if newHash == currentHash:

            continue

            print("hay un nuevainformacion")

            crawler.extraer_datos()

        else:

            print("something changed")

            print(currentHash).text

            print(datetime.datetime)

            response = urlopen(url).read()

            currentHash = hashlib.sha224(response).hexdigest()

            time.sleep(30)

            continue

    except Exception as e:

        print("error")
```

Las tareas de la funcionalidad de búsqueda de información se realizan de forma automática utilizando los datos configurados por el usuario en la parte de configuración de búsqueda,

estos datos son utilizados a través de todo el proceso de extracción de información. En esta fase y en las siguientes el usuario no interviene en el proceso de recolección de información. la Figura 3 muestra el proceso de la búsqueda de información en el cual podemos observar el flujo en el proceso de las tareas realizadas dentro de esta fase.

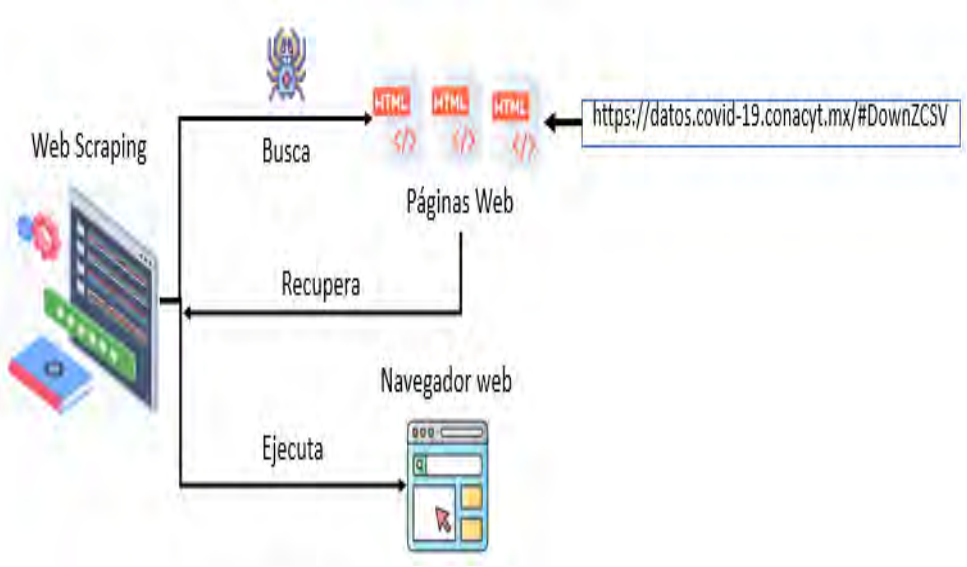

Figura 3. Diagrama de la búsqueda de información

4.2.3. Recolección de información

La tercera funcionalidad del sistema *BDScraping* (Figura 1) tiene como objetivo iniciar el proceso de ubicación y recolección de los elementos a recuperar. El sistema inicia un análisis de la estructura HTML recuperada de la página Web realizando una navegación Web en el sitio que se encuentra abierto en el navegador.

La navegación que realiza el *BDScraping* es una emulación de lo que se realizaría manualmente en una búsqueda de información. las rutas *Xpath* que son proporcionadas por el usuario en la configuración de búsqueda se utilizan para pasar de nodo a nodo dentro de la estructura HTML ubicando los elementos a los que apunta cada una de las rutas *Xpath* y verificando si existen elementos primos que pertenezcan al mismo tipo dentro de un bloque de código de estructura HTML, de esta forma el sistema identifica los elementos a descargar.

Las rutas *Xpath* permiten al sistema realizar interacciones con la página Web, estas tienen tres posibles acciones para realizar dentro de la página Web.

La primera acción: el sistema realiza un clic encima de un elemento del sitio Web lo cual genera una acción (despliegue de menú, selección de dato, abrir un enlace) dentro del sitio Web a lo cual obtendremos una respuesta. El código que se muestra en Tabla 11 a continuación permite ver como el sistema realiza una petición de clic dentro de un sitio Web.

Tabla 11. Ejemplo de búsqueda de Xpath

```
dato = Web DriverWait(driver, 10).until(EC.element_to_be_clickable((By.XPATH,
'//*[@id="bPvM1"]/a')))

dato.click()
```

La segunda acción: el sistema realiza un *scrolling* en la página Web cargando así nueva información en la página Web, la cual sólo se puede ver al realizar esta acción dentro del sitio Web.

La tercera acción: el sistema recupera un atributo de algún elemento de la estructura HTML y lo guarda dentro de una lista que se utiliza como marcador para los elementos a descargar. El código que se muestra en la Tabla 12 presenta como el *BDScraping* realiza esta acción.

Tabla 12.Ejemplo de recuperación de link de un archivo

```
dato5 = Web DriverWait(driver, 20).until(EC.element_to_be_clickable((By.XPATH,
'//    *[@id    ="datos_abiertos_table"]/table/tbody/tr['    +    str(i)    +
']/td[3]/a'))).get_attribute("href")

vr.imprimirD(dato5)

print(dato5)

listalink.append(dato5)
```

La navegación automática por el sitio Web permite al *BDScraping* tener la seguridad de que no se excluyó ningún elemento de la página Web. Una vez terminado el análisis y la navegación el *BDScraping* obtiene una lista con todas las rutas de descarga de los archivos que se descargaran del sitio Web. Esta lista es temporal por lo cual al terminar el análisis se envía una copia al método de descarga para que este pueda ocupar estos datos para realizar sus operaciones. En la Figura 4 se presenta el diagrama de la recolección de la información.

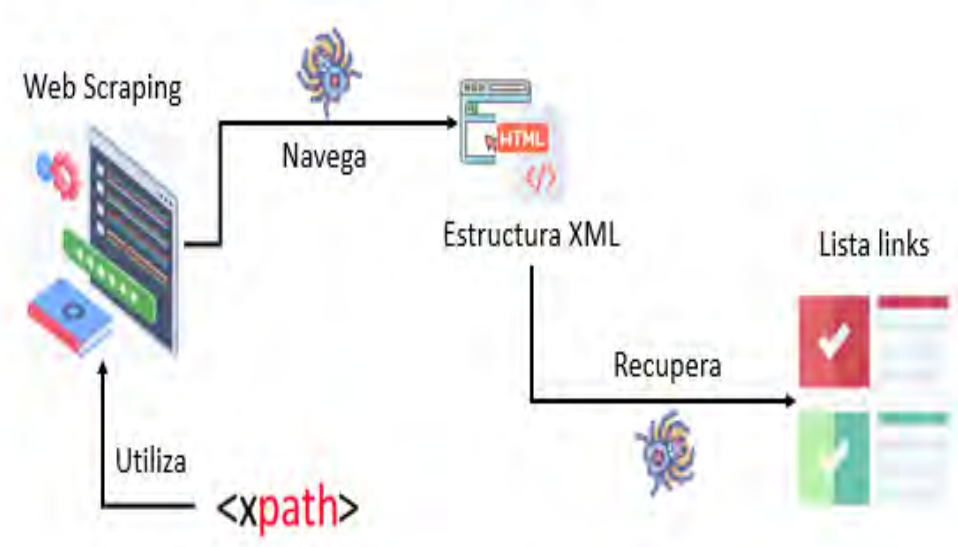

Figura 4.Diagrama de la recolección de la información

4.2.4. Descarga de información

La cuarta funcionalidad del *BDScraping* (Figura 1) tiene como objetivo iniciar el proceso de descarga de los elementos identificados del sitio Web. El sistema utiliza la lista de links que se generó en el método de recolección de información para realizar peticiones de descarga al servidor donde se encuentran alojados los archivos que se descargaran con el sistema.

Si se obtiene una respuesta positiva por parte del servidor, el *BDScraping* recupera el archivo y lo abre para revisar el contenido de esta forma se corrobora que el archivo tenga información dentro, por último, el sistema guarda la información en formato CSV en la ruta de la base de datos especificada por el usuario. La Tabla 13 presenta un fragmento de cómo se realiza este proceso.

Tabla 13.Ejemplo de descarga de archivo

```
def descargaArchivo(listalink):

print('descargando archivos')

    for d in listalink:

        r = requests.get(d, allow_redirects=True)

        file_name  =  'C:/Users/MIGAJAS  PC/Downloads/serendipia/'  +  d[46:-4]+
'.csv'

        output = open(file_name, 'wb')

        output.write(r.content)

        output.close()

        print(file_name)
```

La información obtenida es pasada al formato adecuado y almacenada dentro de la base de datos que se ingresó en la configuración de búsqueda. cada elemento descargado se guarda en un nuevo registro creando el conjunto de datos unificado. El proceso de descarga se realiza por cada elemento dentro de la lista de link. En la Figura 5 se puede observar el proceso de descarga de la información.

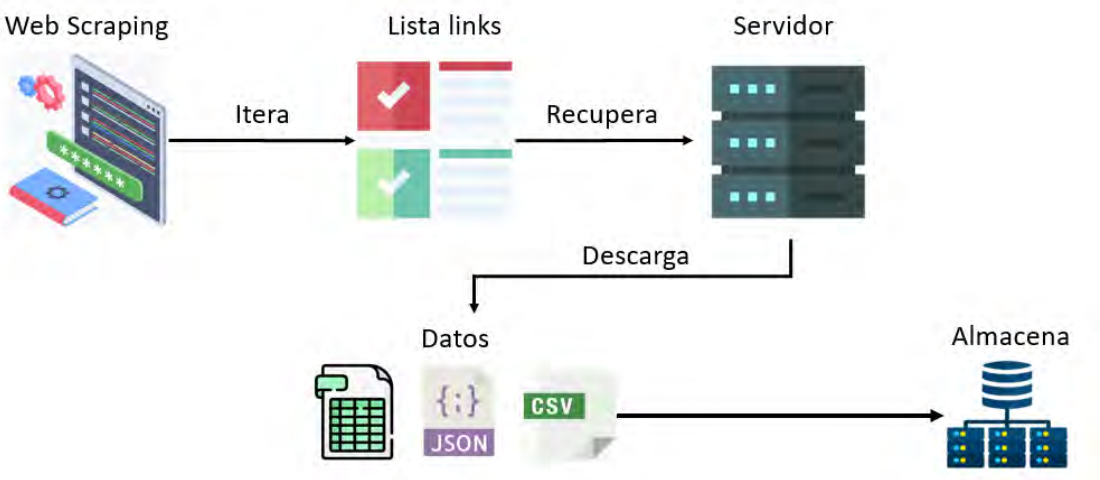

Figura 5.Diagrama de la descarga de la información

4.2.5. Monitoreo de información

La quinta funcionalidad del BDScraping (Figura 1. Modelo SADT del funcionamiento del sistema BDScraping) tiene como objetivo realizar el monitoreo de los sitios Web de los cuales se recolecto información. El sistema realiza requerimientos de acceso a los servidores donde se encuentran alojadas las páginas Web que se van a monitorizar, estas solicitudes de acceso permiten recuperar la estructura HTML del sitio Web, una vez recuperada esta información el sistema realiza un mapa de la estructura HTML. La Tabla 14 presenta un fragmento de cómo se trabaja este proceso.

Tabla 14.Ejemplo de generación de lectura y creación de mapa de sitio Web

```
Response = urlopen(url).read()

currentHash = hashlib.sha224(response).hexdigest()

time.sleep(30)

response = urlopen(url).read()

newHash = hashlib.sha224(response).hexdigest()

if newHash == currentHash: continue
```

La recuperación de información de un sitio Web genera un mapa al inicio del proceso de recuperación. El mapa que se genera en esa extracción de datos se compara con el mapa

generado en esta fase, así, se detectan los cambios existentes en la información destacando las adiciones, actualizaciones y eliminaciones de información.

La fase de monitoreo detecta los cambios en los datos del sitio Web y ejecuta un nuevo proceso de descarga de información tomando las características configuradas por el usuario y reemplazando el mapa del sitio Web que se generó en una primera descarga, actualizando de esta forma la información. En caso de una eliminación del sitio Web o un cambio dentro de la estructura HTML para su distribución el sistema solicitara una nueva configuración de búsqueda de información. La Figura 6. Diagrama del monitoreo del sitio Web realizado por el BDScraping muestra el flujo del proceso de monitoreo del sistema BDScraping.

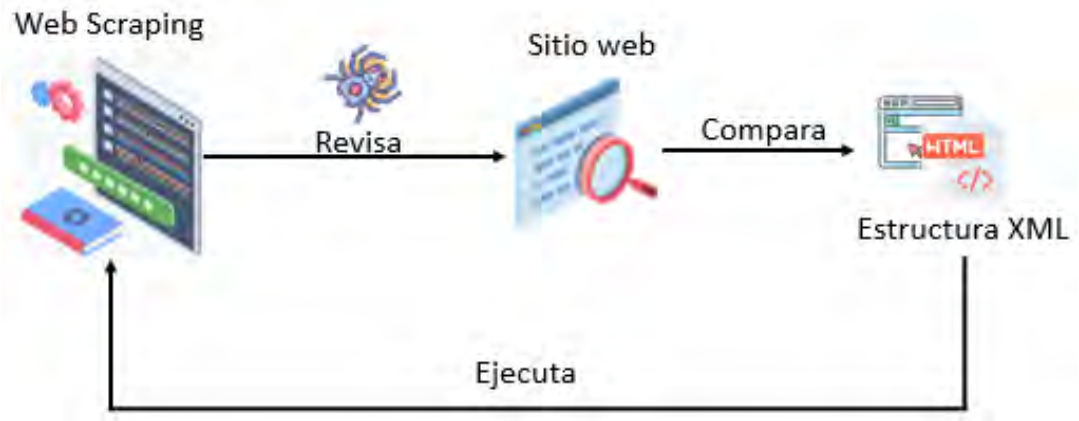

Figura 6. Diagrama del monitoreo del sitio Web realizado por el BDScraping

Capítulo 5

Experimentación y Resultados

5.1. Métricas de evaluación

El análisis para la evaluación del sistema BDScraping, el cual permite recuperar información Web de forma automatizada, requiere de métricas con las cuales se pueda medir y contrastar los resultados que se obtengan en cada extracción.

Las métricas más utilizadas para la evaluación de un sistema de recuperación de información se centran en determinar qué tan preciso y correcto es el resultado obtenido por parte del sistema partiendo de la importancia de la información recuperada.

Las métricas utilizadas en este trabajo e investigación son la de exhaustividad, precisión y el valor F1. Estas requieren de la especificación de una relevancia de la información que se busca recuperar para obtener los valores en porcentaje de lo esperado y lo obtenido.

5.2. Caso de estudio

El caso de estudio propuesto para la evaluación de la metodología del *BDScraping* es el COVID19, la cual es una enfermedad respiratoria que se ha propagado por todo el mundo desde el año 2019. Las dependencias de salud han sido los encargados de publicar la cantidad de casos confirmados, casos de defunciones, casos recuperados y la ubicación de los mismos de manera constante a través de los diferentes medios y canales de transmisión de información. Uno de los medios más utilizados para compartir información del COVID19 son las páginas Web de carácter gubernamental y de los servicios de salud, creando así múltiples fuentes de información que pueden ser consultados por investigadores para realizar diferentes estudios científicos referentes al COVID19. La cantidad de información encontrada en páginas Web sobre este tema nos permite tener una cantidad considerable de datos con lo cual se puede realizar diferentes pruebas y mediciones en los resultados que se obtengan.

5.3. Experimentos

Las pruebas realizadas al *BDScraping* se enfocaron en la recuperación de forma automática de datos sobre el COVID19 en México. Se realizó una investigación en donde se buscó sitios Web que publican información de la pandemia provocada por COVID19 en México. La información publicada en estos sitios es de licencia libre y se publica periódicamente por lo cual cualquier usuario puede hacer uso de esta información. Los datos se distribuyen en formatos PDF, CSV o consultas directas a la página y su base de datos. Las pruebas que se realizaron se centran en la recuperación de información sobre los casos positivos, casos negativos, casos sospechosos y las defunciones de pacientes diagnosticados con COVID19. La extracción de esta información permitió generar un conjunto de datos unificados que pueden ponerse a disposición de investigadores que lo requieran. Las búsquedas realizadas

arrojaron 9 sitios Web en los cuales se publica información de la pandemia por COVID-19 en México. En la Figura 7 Página Web con contenido del COVID19 Figura 7, se presenta un sitio Web con información disponible sobre la pandemia del COVID19.

Figura 7 Página Web con contenido del COVID19

De los sitios Web encontrados se seleccionaron 6 de ellos debido a que estos publican una serie de reportes en formatos .CSV los cuales pueden ser descargados, estos registros contienen información de los casos positivos, casos negativos, casos sospechosos y defunciones de forma nacional y estatal. Los periodos de actualización de esta información son de 24 horas en las cuales se publican nuevos datos actualizando la información disponible. Los sitios seleccionados se separaron en dos grupos para realizar las pruebas correspondientes como podemos observar en la Tabla 15 y la Tabla 16.

Tabla 15. Grupo uno de sitios Web para la realización de Web Scraping

Num	Nombre del sitio Web	URL
1	CONACYT	https://datos.covid-19.conacyt.mx/
2	Gobierno Federal	https://coronavirus.gob.mx/
3	Apple	https://covid19.apple.com/

Tabla 16.Grupo dos de sitios Web para la realización de Web Scraping

Núm.	Nombre del sitio Web	URL
1	Human data	https://data.humdata.org/
2	Serendipia	https://serendipia.digital/
3	Gobierno CDMX	https://datos.cdmx.gob.mx/

5.4. Prueba 1 de extracción de información

La prueba 1 de extracción de información tiene como objetivo la recuperación de información sobre la pandemia por COVID19 en México. Esta prueba busca obtener los datos de tres sitios Web que cuentan con información de pacientes de casos positivos, negativos, sospechosos y recuperados, esto con el fin de generar una base de datos única que pueda ser utilizada para el análisis de evolución de la pandemia por COVID19 durante un periodo de tiempo establecido.

El primer grupo de pruebas de extracción consideró tres sitios Web y se configuraron las búsquedas para cada uno de las páginas Web. El primer grupo se formó por las siguientes páginas Web; datos. covid19.conacyt, coronavirus.gob.mx, covid19.apple.com, Cada sitio Web se configuró independiente de los otros, pero compartiendo la misma base de datos, de esta forma se obtuvieron tres configuraciones a los cuales se les ingreso los valores de entrada como se muestra en la Tabla 17.

Tabla 17.Configuración de extracción de grupo uno

Nombre de búsqueda	Base de datos	URL de la página Web	Dominio de búsqueda	Rutas Xpath
Covid conacyt	Covid19	https://datos.covid19.conacyt.mx/#DownZCSV	Datos.covid-19.conacyt.mx	//*[@id="bPvM1"]/a/span[2] //*[@id="bPvM2"]/a/span[2]
Covid gobierno	Covid19	https://datos.gob.mx/busca/dataset/informacion-referente-a-casos-covid-19-en-mexico.	Datos.gob.mx	//*[@id="content"]/a
Covid apple	Covid19	https://covid19.apple.com/mobility	Covid19.apple.com	//*[@id="downloadcard"]/div [2

La configuración de estas búsquedas fue utilizada por el *BDScraping* para recolectar información de los dominios utilizados durante una semana. Cada configuración de búsqueda se ejecutó en el mismo horario corriendo la descarga de la información a las 10:00 de la noche durante los 7 días que se tomaron como rango de extracción. En la Figura 8 se muestra la configuración de la búsqueda.

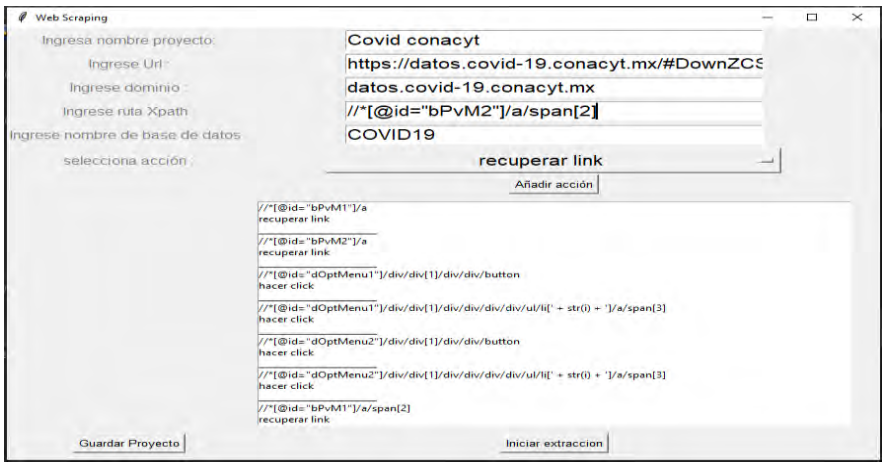

Figura 8. configuración de búsqueda

5.5. Prueba 2 de extracción de información

La prueba 2 de extracción de información tiene como objetivo la recuperación de información sobre la pandemia por COVID19 en México. Esta prueba busca obtener los datos de tres sitios Web que cuentan con información de pacientes de casos positivos, negativos, sospechosos y recuperados y cualquier otro dato que esté relacionado con el COVID19, esto con el fin de generar una base de datos única que pueda ser utilizada para el análisis de evolución de la pandemia por COVID19. Esta extracción de información utiliza el monitoreo de datos.

El segundo grupo de pruebas de extracción consideró tres sitios Web y se configuraron las búsquedas para cada uno de las páginas. El segundo grupo se formó por las siguientes páginas Web; data.humdata.org, serendipia. Digital y datos.cdmx.gob.mx. Cada sitio Web se configuro independiente de los otros, pero compartiendo la misma base de datos, de esta forma se obtuvieron tres configuraciones a los cuales se les ingreso los valores de entrada como se muestra en la Tabla 18.

Tabla 18. Configuración de pruebas dos

Nombre de búsqueda	Base de datos	URL de la página Web	Dominio de búsqueda	Rutas Xpath
Covid humdata	Covid19	https://data.humdata.org/dataset?groups=mex&res_format=CSV&q=covid&sort=if(gt(last_modified%2Creview_date)%2Clast_modified%2Creview_date)%20desc&ext_page_size=25	Data.humdata.org	//*[@id="search-page-results"]/div/ul/li[7]/div/div/div/div[1]/span/div[1]/a
Covid serendipia	Covid19	https://serendipia.digital/covid19-mx/datos-abiertos-sobre-casos-de-coronavirus-covid-19-en-mexico/	serendipia. digital	// *[@id="datos_abiertos_table"]/table/tb ody/tr['+ str(i) +']/td[3]/a
Covid cdmx	Covid19	https://datos.cdmx.gob.mx/group/covid-19	datos.cdmx.gob.mx	//*[@id="content"]/div[3]/div/article/div/ul/li[3]/ul //*[@id="content"]/div[3]/div/article/div/ul/li[12]/ul/li[1]/a

La configuración de estas búsquedas fue utilizada por el sistema *BDScraping* para recolectar información de los dominios utilizados durante una semana. El sistema ejecutó una primera descarga de información y posteriormente quedo en espera de un cambio en la información ejecutando un centinela que revisaba el sitio Web utilizado cada 12 horas en la cual realizaba un mapa de la página Web y lo comparaba con el realizado en la primera extracción. Este proceso lo realizó durante los 7 días que se marcó como periodo de tiempo para este experimento. En la Figura 9.

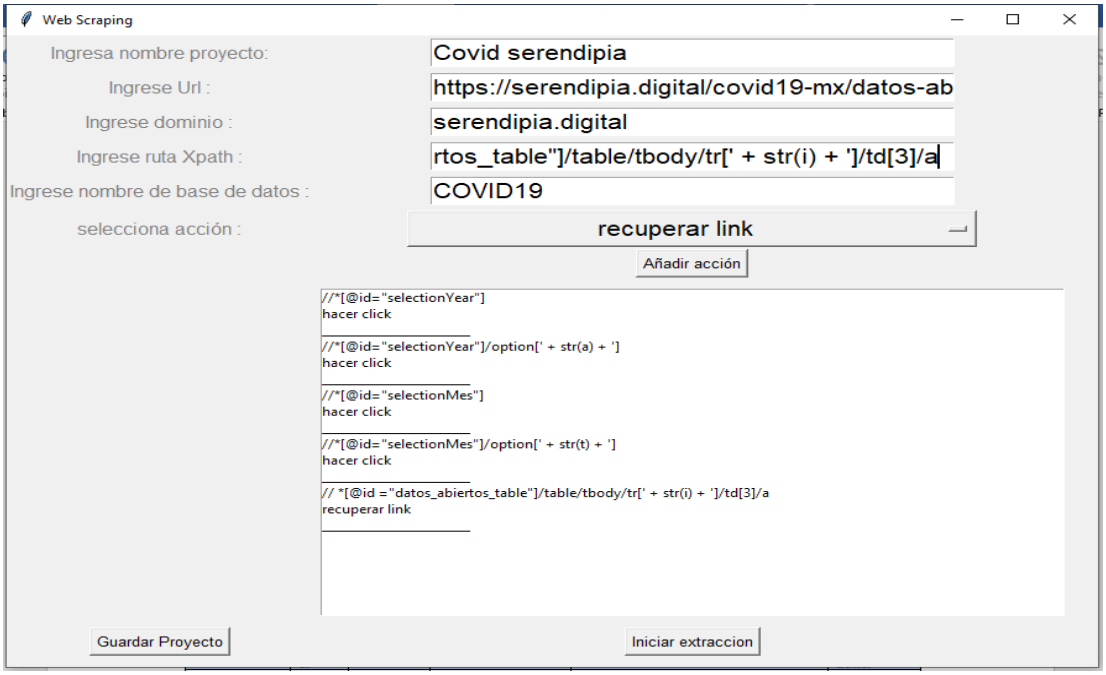

Figura 9. Configuración de la búsqueda

5.6. Resultados de los experimentos

Los resultados obtenidos en la experimentación fueron favorables ya que se logró obtener una cantidad considerable de información sobre el COVID19. Cada búsqueda realizada obtuvo seis datos para cada archivo que se logró obtener de las páginas Web. Los datos que se recuperaron son; título de la información obtenida, descripción de la información obtenida, la fecha de la extracción de la información, la URL de la página Web, el nombre del archivo obtenido. A continuación, se muestran los resultados obtenidos de los dos grupos de experimentos realizados.

El grupo uno de sitios Web cuya configuración se ejecutó durante una semana y con un horario de extracción establecido logro obtener un total de 69 archivos con información sobre COVID19. En la Tabla 19 se observan los resultados de las búsquedas de información del grupo uno.

Tabla 19. Resultados de las búsquedas de información del grupo uno

Título	Descripción	Fecha	URL Página	Archivos	Cantidad archivos
Casos diarios por estado + nacional	Casos confirmados Casos sospechosos Casos negativos Casos defunciones	09-05-2022 – 15-05-2022	ttps://datos.covid19.conacyt.mx/#Down ZCSV	Casos_Diarios_Estado_Nacional_Confirmados_20220309 Casos_Diarios_Estado_Nacional_Sospechosos_20220309 Casos_Diarios_Estado_Nacional_Negativos_20220309 Casos_Diarios_Estado_Nacional_Defunciones_20220309 Casos_Diarios_Municipio_Confirmados_20220309 Casos_Diarios_Municipio_Sospechosos_20220309 Casos_Diarios_Municipio_Negativos_20220309 Casos_Diarios_Municipio_Defunciones_20220309	56
Información referente a casos COVID-19 en México	Base de datos con datos sobre covid en México	09-05-2022 – 15-05-2022	https://datos.gob.mx/busca/dataset/informaci on-referente-a-casoscovid-19-en-m	Datos_abiertos_covid.csv	7
informes de tendencias de movilidad	Datos de movilidad	09-05-2022 – 15-05-2022	https://covid19.apple.c om/mobility	applemobilitytrends-2022-03-09	6

La cantidad de información que se puede obtener de las diferentes páginas Web varía según el tipo de información que se maneje. En algunas páginas Web la información es puesta a disposición de forma seccionada según los índices que la página maneje, por lo cual la cantidad de archivos aumenta o disminuye según como se distribuyan los datos dentro de la página Web.

El grupo dos de sitios Web se ejecutó durante siete días con la utilización de un centinela el cual detecto los cambios de la información dentro de las páginas Web. El sistema ejecutó las descargas de información cada que se detectó un cambio obteniendo una cantidad de 1098 archivos en el periodo de tiempo establecido. En la Tabla 20 se observan los datos obtenidos en la extracción de información del grupo dos de sitios Web.

Tabla 20. Resultados de la extracción de información del grupo dos de sitios Web

Título	Descripción	Fecha	URL Página	Archivos	Cantidad de archivos
México: Reported COVID-19 Cases	Reportes de casos covid	09-05-2022 – 15-05-2022	https://data.humdata.org/dataset/mexico-reported-covid-19-	MXCOVID.csv	60
Datos abiertos de COVID-19 en México	Casos positivos Casos sospechosos	09-05-2022 – 15-05-2022	https://serendipia.digital/covid19-mx/datos-abiertos-sobre-casos-de-coronavirus-covid-19-en-mexico/	covid-19-mexico-201231.csv covid-19-mexico-sospechosos-201231.csv	1010
Históricos casos COVID19	Histórico de casos covid	09-05-2022 – 15-05-2022	https://datos.cdmx.gob.mx/group/covid-19?q=covid19&sort=score+desc%2C+metadata_modified+desc	Histórico Mensual de casos Covid19.csv	28

5.7. Evaluación de los resultados

La evaluación de los resultados obtenidos de los proyectos configurados para el sistema de *Web Scraping* se realizaron aplicando las métricas de precisión, exhaustividad y el valor F1.

Las métricas requieren de un número de elementos esperados que sirve como valor de referencia para la comparativa de lo recolectado de forma automática a lo recolectado manualmente, este valor es tomado como el número de datos esperados de una descarga de información Web. A continuación, se enlistan los pasos que se siguieron para la evaluación:

1. El proceso de búsqueda, descarga y almacenamiento se realizó de forma manual para obtener el número de datos a esperar en una extracción de información de forma automática.
2. Los proyectos de *Web scraping* fueron configurados ingresando los parámetros de entrada para cada uno de los proyectos.
1. 3.S e ejecutaron todos los proyectos configurados durante una semana en la cual se obtuvo información de diferentes páginas Web.
3. Los datos recuperados fueron revisados para verificar que contenían la información sobre el COVID19. Los índices que se encontraron en los datos correspondieron a casos positivos, casos negativos, casos sospechosos y defunciones de pacientes que se buscaba recuperar.
4. Las métricas de evaluación fueron aplicadas para obtener los porcentajes de precisión y exhaustividad de la información recuperada en comparación a la esperada. En la Tabla 21y la Tabla 22 se presentan los resultados obtenidos de las métricas de precisión, exhaustividad y el valor F1.

La evaluación de los resultados de las pruebas de funcionamiento obtuvo resultados positivos como se muestran en las tablas Tabla 21 y Tabla 22. Los números que se muestren en esas tablas nos permiten aseverar que los resultados son favorables ya que se recuperaron las cantidades de archivos similares a las esperadas en el análisis inicial de los sitios Web.

Tabla 21. Resultados de las pruebas de extracción con el sistema BDScraping

Proyecto de *Web scraping*	Descarga manual de datos relevantes (semana)	Descarga con *Web scraping* datos relevantes	Datos descargados no relevantes	Métrica de precisión	Métrica exhaustividad
Covid conacyt	56	56	2	96%	100%
Covid gobierno	7	7	0	100%	100%
Covid apple	7	6	1	75%	85%
Covid humdata	63	60	4	89%	95%
Covid serendipia	1014	1010	1	99%	99%
Covid cdmx	35	28	2	75%	80%

Tabla 22. Resultados de la aplicación de las métricas de evaluación

Proyecto de *Web scraping*	Valor precisión	Valor exhaustividad	Valor F1
Covid conacyt	96	100	97
Covid gobierno	100	100	100
Covid apple	85	75	79
Covid humdata	89	95	91
Covid serendipia	99	99	99
Covid cdmx	75	80	77

Capítulo 6

Conclusiones

6.1. Conclusiones

Se obtuvieron resultados favorables en la recuperación de información almacenada en páginas Web utilizando del sistema desarrollado en esta tesis, denominado *BDScraping,* El sistema permitió recuperar información relevante para el caso de estudio COVID19, con índices de casos positivos, casos negativos, casos sospechosos y defunciones. En este sentido se logró recuperar un conjunto de datos significativo, lo cual permitió unificar así información sobre el COVID19 en sitio único.

Las métricas de precisión obtuvieron valores altos en los datos obtenidos de las extracciones, aun cuando se recolectaron datos con valores nulos o vacíos o con información que tiene que ver con el caso de estudio del COVID19, pero que queda fuera de los índices estudiados. La métrica de exhaustividad tuvo mejores resultados, puesto que se logró recolectar un número cercano al valor de referencia que se tenía como valor de referencia.

Los resultados obtenidos permiten afirmar que la información publicada en páginas Web puede ser recuperada de forma automática y generar de esta forma un conjunto de datos disponible para diferentes fines. La descarga de esta información requiere por supuesto de una correcta configuración por parte del usuario. Este debe ingresar los parámetros iniciales de nombre de proyecto, dominio de búsqueda, URL del sitio Web, nombre de la base de datos y las rutas Xpath con las acciones que estas realicen para lograr una correcta extracción de la información de lo contrario el sistema detectará como una mala configuración y puede haber fallas dentro de los elementos recuperados.

6.2. Contribuciones

La principal contribución de este trabajo de tesis es un sistema informático que permite obtener información de los sitios Web elegidos por los usuarios. Este sistema solicita una configuración de búsqueda de información la cual se realiza una única vez y sólo se modifica en caso de una actualización de la página Web o la necesidad de obtener nuevos datos del sitio Web.

La aplicación desarrollada utiliza sistemas de *Web Scraping* con lo cual se puede realizar extracciones tanto lateralmente pasando de página a página o realizar una obtención de datos de forma vertical, ingresando a los diferentes enlaces de navegación hasta llegar a la información de interés.

El sistema está pensado para obtener únicamente documentos o archivos dentro de la página Web, sin embargo, el sistema puede ser modificado para obtener cualquier tipo de información que se plantee recuperar. Para esto, se requiere de los datos de configuración para obtener la información de forma correcta.

6.3. Trabajos futuros

La recolección de información es parte fundamental en la toma de decisiones o realización de trabajos de investigación. El sistema BDScraping desarrollado en esta tesis permite obtener cierta información de forma automatizada, logrando obtener y almacenar de forma adecuada los datos que se quieran ocupar.

Un enfoque interesante para la mejora del BDScraping es la implementación de un lector de Xpath en tiempo real, el cual permita recuperar la dirección de la ruta Xpath dando un clic sobre del elemento a recuperar. En este caso el sistema sería el responsable de realizar el recorrido de la página Web y obtendría los datos necesarios para una extracción de información correctamente.

6.4. Logros obtenidos

El primer logro obtenido durante el desarrollo de este trabajo de investigación fue la redacción y publicación de un artículo científico en el XIV Congreso Mexicano de Inteligencia Artificial COMIA 2022. El artículo lleva por nombre obtención automática de bases de datos en la Web: utilizando técnicas de *Web Scraping*, donde se presenta una parte de lo realizado con la creación del BDScraping.

El segundo logro fue la presentación de la conferencia obtención automática de bases de datos en la Web: utilizando técnicas de *Web Scraping*, en el XIV Congreso Mexicano de Inteligencia Artificial COMIA 2022. La cual se realizó de forma virtual en la ciudad de Oaxaca.

Referencias

Aldana, J. S. (2002). Los sitios web como estructuras de información. *Bibliotecnología y ciencias de la información, 3*(12).

Barnard, A., Delgado, A., & Voutssás, J. (2016). Introducción al Cómputo en la Nube. *Cuadernos Digitales de Archivística*. Ciudad de México: Archivo General de la Nación.

Bass, L., Clements, P., & Kazman, R. (2012). *Software Architecture in Practice, Third Edition.* Boston: Addison-Wesley Professional.

Bessis, N., & Dobre, C. (2014). Big Data and Internet of Things: A Roadmap for Smart Environments. *Studies in Computational Intelligence (SCI, volume 546).* Springer.

Bo, Z. (2017). Web Scraping. *Encyclopedia de big data* (págs. 1-3). Corvallis: Springer.

Cai, L., & Zhu, L. (2015). The Challenges of Data Quality and Data Quality Assessment in the Big Data Era. *Data Science Journal, 14*, 2.

Carrion, J. (2010). *Diferencia entre dato, información y conocimiento.* Recuperado en Enero de 2022, http://iibi.unam.mx/voutssasmt/documentos/dato%20informacion%20conocimiento.pdf

Cecilia, A. G. (2014). Las tecnologías de la información y la comunicación (TIC) un instrumento para la investigación. *Investigaciones Andina, 16*(29), 997-1000.

Dowling, M. (2015). *Guzzlephp.org*. Recuperado el Enero de 2023, https://docs.guzzlephp.org/en/stable/

fabpot. (2020). *goutte*. (Goutte a simple PHP web scraper). Recuperado el 20 de enero de 2021, https://github.com/FriendsOfPHP/Goutte

Glez-Peña, D., Lourenc¸o, A., López Fernández, H., Reboiro Jato, M., & Fdez Riverola, F. (2013). Web scraping technologies in an API world. *Briefings in bioinformatics, 15*(5), 788-797.

González Jaimez, C. (2015). *Programacion de web estático.* México: Universidad Autónoma Metropolitana Ed., Unidad Cuajimalpa.

González Moreno, J. (2001). JavaScript: diseño de páginas Web dinámicas y algo más. Cuesta Morales, & J. L. Martínez Orge Ed., *Diseño gráfico y desarrollo de aplicaciones para la web* (págs. 11-26). España: Aica.

Hernández, A. T., Gómez Vazquez, E., Berdejo Rincón, C. A., Hernández, A. T., Montero García, J., & Calderón Maldonado, A. (2015). Metodologías para análisis político utilizando Web Scraping. *Research in Computing Science 95*, 113-121.

import.io. (s.f.). *import.io*. Recuperado el 05 de Febrero de 2023, de https://www.import.io/

Martinez Rebollar, A., Pech May, F., Estrada Esquivel, H., & Pedroza Landa, E. (2015). CrawNet:un Crawler para obtener Información de Recursos Multimedia de la Web Superficial y Oculta. *Lámpsakos, 13*, 39-50.

Martínez, R., Rodríguez, R., vera, P., & Parkinson, C. (2019). Análisis de técnicas de raspado de datos en la web aplicado al Portal del Estado Nacional Argentino. *XXV Congreso Argentino de Ciencias de la Computación (CACIC).* Cordoba: UniRío Editora.

Mendoza , M. (2011). Minería de datos en la web. En F. Cacheda, J. F. Huete Guadix, & J. Fernández Luna, *Recuperación de información un enfoque práctico y multidisiplinar* (págs. 613-648). España: Ra-Ma.

Microsoft. (s.f.). *learn.microsoft*. (microsoft) Recuperado el Febrero de 2023, de https://learn.microsoft.com/es-es/dotnet/standard/base-types/regular-expression-language-quick-reference

Moreno, J. P. (2014). Una aproximación a big data. *Revista de Derecho de la Universidad Nacional de Educación a Distancia, 14*, 471.

Parra, D. (2016). *parra.sitios.ing.uc.cl.* Recuperado el Enero de 2023, de https://dparra.sitios.ing.uc.cl/

Penman, R. B., Baldwin, T., & Martínez, D. (2010). Web Scraping Made Simple with SiteScraper. *Computer Science*.

Ram Sharan, C., Santosh, P., & Sadhu Ram, B. (2017). Cloud Based Web Scraping for Big Data Applications. IEEE International Conference on Smart Cloud, (págs. 138-143). New York, NY, USA.

Richardson, L. (s.f.). *Beautiful Soup Documentation*. Recuperado el 05 de Febrero de 2023, de https://www.crummy.com/software/BeautifulSoup/bs4/doc/

Sagiroglu, S., & Sinanc, D. (2013). Big data: a review. *International Conference on Collaboration Technologies and Systems (CTS).* San Diego: IEEE.

Saurkar, V. A., Pathare, K. G., & Gode, S. A. (2018). Una descripción general de las técnicas y herramientas de web scraping. *Revista internacional sobre la revolución futura en ciencias de la computación e ingeniería de la comunicación, 4*, 363-367.

Scrapy.org. (s.f.). *scrapy*. Recuperado el Febrero de 2023, de https://scrapy.org/

Sirisuriya, D. S. (2015). A Comparative Study on Web Scraping. *8th International Research Conference, KDU.* Sri Lanka.

Uzun, E. (2020). A Novel Web Scraping Approach Using the Additional Information Obtained From Web Pages. *IEEE Access, 8*, 99. doi:10.1109/ACCESS.2020.2984503

Vargas Pérez, A. (2019). Herramienta para recopilar la información en las noticias publicadas en los sitios web de internet. *Serie Científica de la Universidad de las Ciencias Informáticas, 10(5)*, 14-24.

Vargiu, E., & Urru, M. (2013). Explotación del web scraping en un enfoque de publicidad web basado en filtros colaborativos. *Journal for the Artificial Intelligence Specialists, 2(1)*.

Villanueva Rodriguez, U. J. (2019). *Investigación y Desarrollo de Técnicas de Scraping.*: Universidad de Alcalá. España.