Pontificia Universidad Católica del Ecuador | SEDE ESMERALDAS

FACULTAD DE ADMINISTRACIÓN

ESCUELA DE SISTEMAS Y COMPUTACIÓN

TESIS DE GRADO

TÍTULO: SOFTWARE PARA WEB SCRAPING DESDE LAS APIS DE REPOSITORIOS DE CÓDIGO

PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS Y COMPUTACIÓN

LÍNEA DE INVESTIGACIÓN

PROGRAMACIÓN Y DESARROLLO DE SOFTWARE

AUTOR

WILMER FABIÁN PIANCHICHE DELACRUZ

ASESOR

ING. JAIME SAYAGO HEREDIA (MGT.)

Esmeraldas, 2020

Tesis de grado aprobada luego de haber dado cumplimiento a los requisitos exigidos por el reglamento de grado de la PUCE Esmeraldas, previo a la obtención del título de INGENIERO DE SISTEMAS Y COMPUTACIÓN.

TRIBUNAL DE GRADUACIÓN

Título: "SOFTWARE PARA WEB SCRAPING DESDE LAS APIS DE REPOSITORIOS DE CÓDIGO"

Autor: PIANCHICHE DELACRUZ WILMER FABIAN

Mgt. Jaime Sayago Heredia f.-……...
Asesor

Mgt. Evelin Flores García f.-……...
Lector N° 1

Mgt. Wilson Chango Sailema f.-……...
Lector N° 2

Mgt. Xavier Quiñónez Kú f.-……...
Director de Escuela

Ab. Alex David Guashpa f.-……...
Secretario General PUCE Esmeraldas

Esmeraldas …. de …… de 2020

DECLARACIÓN DE AUTORÍA

Yo, PIANCHICHE DELACRUZ WILMER FABIAN, portador de la cédula de identidad No. 0804161230, declaro que la presente investigación enmarcada en el actual trabajo de tesis es absolutamente original, auténtica y personal.

En tal virtud, manifiesto que el contenido, las conclusiones y resultados, y los efectos legales y académicos que se desprendan del trabajo propuesto de investigación y luego de la redacción de este documento son y serán de mi única y exclusiva responsabilidad legal y académica.

Pianchiche Delacruz Wilmer Fabian,
C.I.: 0804161230

CERTIFICACIÓN

Yo, Mgt. Jaime Sayago Heredia, docente investigador de la PUCE Sede Esmeraldas, certifico que:

El trabajo de grado realizado por PIANCHICHE DELACRUZ WILMER FABIAN bajo el título "SOFTWARE PARA WEB SCRAPING DESDE LAS APIS DE REPOSITORIOS DE CÓDIGO" reúne los requisitos de calidad, originalidad y presentación exigibles a una investigación científica, y que además han sido incorporadas al documento final las sugerencias realizadas, en consecuencia, está en condiciones de ser sometido a la valoración del Tribunal encargado de juzgarlo.

Y para que conste a los efectos oportunos, firmo la presente en Esmeraldas, el ….. de ……. de 2020.

Mgt. Jaime Sayago Heredia
Asesor

DEDICATORIA

El presente trabajo investigativo le dedico principalmente a:

Dios

por ser el inspirador, y darme fuerza y por haberme permitido llegar hasta este punto y haberme dado salud para lograr mi objetivo, además, de su infinita bondad y amor.

Mis Padres

por su amor, trabajo y sacrificio en todos estos años, gracias a ustedes he logrado llegar hasta aquí y convertirme en lo que soy. Ha sido el orgullo y el privilegio de ser su hijo, son los mejores padres.

Al igual que todos mis logros pasados y futuros están completamente dedicados a mis padres, quienes, con su comprensión, apoyo, sabiduría y sobre todo amor, han sabido formarme durante toda mi carrera estudiantil, y, sobre todo, durante toda mi vida.

Mis Hermanas / os

por estar siempre presentes, acompañándome y por el apoyo moral que me brindaron a lo largo de esta etapa de mi vida.

Mis amigos - Compañeros

que me han apoyado y han hecho que el trabajo se realice con éxito en especial a aquellos que me abrieron las puertas y compartieron sus conocimientos.

A ustedes les dedico este logro.

Tsaave.

AGRADECIMIENTOS

Agradecer a Dios por bendecirme la vida, por guiarme a lo largo de mi existencia, ser el apoyo y fortaleza en aquellos momentos de dificultad y de debilidad.

Gracias a mis padres: Seferino y María,
por ser los principales promotores de mis sueños, por confiar y creer en mis expectativas, por los consejos, valor y principios que me han inculcado.
A mis hermanas (os) y a toda mi familia,
por el apoyo constante y el cariño incondicional brindado
a cada momento durante cada día.
Ustedes son el mayor apoyo en mi vida

A mis amigos, por alentarme cuando más lo necesito, por extender su mano en momentos difíciles y por estar siempre presentes dándome risas y alegrías.
Me faltarían páginas para describir todas las locuras y
los buenos momentos a su lado.

A los docentes de la Escuela de Sistemas de la PUCE Sede Esmeraldas por haber compartido sus conocimientos a lo largo de mi formación profesional, en especial a mi asesor y lectores, por servir de guía durante la realización de este trabajo.

¡Gracias infinitas!

¡Ayu demangatasai!

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES

ÍNDICE DE TABLAS

RESUMEN

La presente investigación fue realizada con el propósito de desarrollar un software para extraer información de forma automática desde la API de un repositorio de código utilizando la técnica de web scraping. El repositorio de código estudiado o usado en este proyecto es el GitHub debido a su amplia popularidad gracias a los distintos desarrolladores de diversas partes del mundo. Con respecto al tema se llevó a cabo una serie de investigaciones bibliográfica o documental en principales bibliotecas científicas virtuales con el fin de establecer definiciones sobre el tema en estudio, de la misma manera sus características, las distintas formas y herramientas disponibles para esta técnica, haciendo hincapié en sus posibles aplicaciones en distintos ámbitos. Además, de cada una de las herramientas obtenidas se realizó tablas de comparación para así poder elegir las más convenientes. Con los datos obtenidos, se pudo estudiar generando conocimientos a través de análisis de estos datos, asimismo el estudio permitió conocer los proyectos o repositorios más activos o con más contribuidores de GitHub.

En este sentido, la aplicación fue desarrollada con el lenguaje de programación Python bajo el patrón de diseño MTV (Model Template View), utilizando como marco de desarrollo (Framework) Django y se utilizó base de datos NoSQL como MongoDB y como editor de código (IDE) se utilizó PyCharm en su versión Community. Con todo ello, la aplicación es capaz de soportar grandes cantidades de información, desde su recolección (Web Scraping desde la API), almacenamiento, hasta su posterior consulta o búsqueda desde la vista hacia la base de datos. Para determinar la factibilidad de Web Scraping como herramienta para extraer información de forma automática aplicada en GitHub fue necesario realizar diferentes pruebas según las distintas técnicas disponibles, es decir, las pruebas se realizaron a nivel de librerías que son ajenos a la plataforma, y asimismo se realizó a nivel del API, siendo este el objetivo principal del estudio, por lo cual, a nivel de librerías o framework como Scrapy, simplemente se utilizó con fines prácticos, no obstante, en este estudio también se describe sobre esta poderosa librería para el ámbito de Web Scraping.

Palabras claves

Raspado web	**API**	**Repositorio de código**	**GitHub**

ABSTRACT

The research was conducted for the purpose of developing software to extract information automatically from the API of a code repository using the web scraping technique. The code repository studied or used in this project is GitHub due to its wide popularity thanks to the different developers from different parts of the world.. With regard to the topic, a series of bibliographic or documentary research was carried out in main virtual scientific libraries in order to establish definitions on the subject under study, in the same way its characteristics, the different forms and tools available for this technique, emphasizing its possible applications in different fields. In addition, comparison tables were made for each of the tools obtained to choose the most convenient ones. With the data obtained, it was possible to study generating knowledge through analysis of this data, as well as the study allowed to know the most active projects or repositories or with more Contributors of GitHub.

In this sense, the application was developed with the Python programming language under the DESIGN (Model Template View) design pattern, using As a Django framework (Framework) and NoSQL database was used as MongoDB and as a code editor (IDE) PyCharm was used in its Community version. With all this, the application can support large amounts of information, from its collection (Web Scraping from the API), storage, to its subsequent query or search from the view to the database. To determine the feasibility of Web Scraping as a tool to extract information automatically applied on GitHub it was necessary to perform different tests according to the different techniques available, that is, the tests were carried out at the level of libraries that are outside the platform, and it was also carried out at the API level, this being the main objective of the study, so, at the level of libraries or framework such as Scrapy, it was simply used for practical purposes, however, this study also describes this powerful library for the field of Web Scraping.

Keywords

Web scraping	**API**	**Code repository**	**GitHub**

INTRODUCCIÓN

Presentación de investigación

La presente investigación, estudia en primera instancia los conceptos relacionados a las diferentes tecnologías que existen en la actualidad que son considerados como herramientas de desarrollo, herramientas para la automatización para la obtención de datos de la web, partiendo con las definiciones, hasta llegar a determinar cuál es el más conveniente según los requerimientos, además de la importancia que tiene cada uno en la actualidad tanto para las empresas, así como para la educación y el desarrollo del software. En este contexto, se presenta una primera idea del conjunto de herramientas llamada Web Scraping, sobre la cual se definen sus características, tipos, ventajas y desventajas, el porqué de su creación y la manera en la cual funcionan cada una de sus partes, asimismo hay que indicar también que el software ScrapGit está desarrollado bajo un único lenguaje de programación que es el Python.

Asimismo, se estudian los principales repositorios de código, principalmente, se enfoca más en GitHub, se incluyen los distintos lenguajes de programación y sus respectivas bibliotecas enfocadas en el campo de Web Scraping con el fin de establecer la herramienta más adecuada para realizar el trabajo.

El presente estudio se encuentra estructurado en cinco capítulos:

Previo a los capítulos a referir, se enfoca el problema de investigación a través de descripción del contexto de investigación, la formulación del problema, posibles causas, objetivos de investigación, siendo éstas generales y/o específicas, a los cuales se les dará respuesta a través de la investigación, también abarca la justificación. Es así como:

En el Capítulo I, se describen los antecedentes de la investigación, y la fundamentación teórica: aportación de las diferentes fundamentaciones: Teórica Filosófica, Developer, Artículos Científicos, Fundamentación Legal, y finalmente las definiciones del glosario de términos relevantes en el contexto de esta investigación.

Con el propósito de tener una mejor idea sobre el tema es importante tener un enfoque holístico de qué es el web scraping y su importancia en las empresas, en la

educación, en los proyectos, ventajas que brinda su aplicación en el proceso de recolección de datos.

El Capítulo II se refiere a todos los instrumentos utilizados para la investigación: métodos de investigación, técnicas e instrumentos de investigación, población y muestra de estudio, descripción y validación del instrumento, técnicas de procesamiento y análisis de datos. También incluye normas éticas.

El Capítulo III principalmente presenta resultados de la investigación.

El Capítulo IV presenta la discusión de resultados de la investigación en base los objetivos planteados, el análisis de la información obtenida en base los antecedentes del estudio.

En el Capítulo V se redactan las conclusiones y recomendaciones en base a los objetivos planteados al inicio de la investigación. Además, se presenta las referencias bibliográficas bajo el estándar IEEE.

Planteamiento del problema

La evolución de la tecnología web tiene un efecto a nivel mundial [1], reflejando un aumento en la dependencia de la tecnología web de la sociedad actual llegando a ser parte de la vida cotidiana. We Are Social y Hootsuite, en el informe global digital 2019, revelan que los cibernautas de la web crecen, en términos medios, por encima de 1 millón de nuevos internautas a diario. Según este informe, actualmente existen en el mundo, un total de 5.11 mil millones de usuarios sólo a través de dispositivos móviles, incluso 100 millones (2%) hasta el año pasado. Englobando así unos 4.39 mil millones de usuarios de Internet sólo en el 2019, lo que significa un incremento de 366 millones (9%) frente a enero de 2018. En el mismo contexto, con respecto a las redes sociales hubieron 3.48 mil millones de usuarios en ese año, generando así un crecimiento a nivel mundial un total de 288 millones (9%). En este sentido, unos 3 260 millones de personas utilizaron redes sociales a través de sistemas o plataformas móviles en enero de 2019, lo que significa un incremento de 297 millones de nuevos cibernautas, con todo ello se entiende que se ha generado un crecimiento anual de más del 10% [2]. Asimismo, UIT (Unión Internacional de Telecomunicaciones) que es la entidad competente de las Naciones Unidas en el ámbito de las tecnologías de la información y la comunicación (TIC), estimaba que, al término del año 2018, el 51,2% de la población a nivel global, en otras palabras, 3 900 millones de personas, estarían utilizando Internet [3].

De esta manera ha impulsado a las instituciones, entidades y a las industrias a la búsqueda de respuestas que permitan brindar un servicio optimizado y de calidad, por ende, mayor disponibilidad, rendimiento y escalabilidad. Los llamados servicios web que son un tipo de programa que bridan los proveedores de servicios, las cuales son ofrecidos por medio de la web, convirtiendo así en un componente fundamental en la incorporación de sistemas de distintos entornos así como lenguajes de programación y tecnologías [1].

En este contexto, se expone un caso particular sobre las enfermedades tropicales que ocurren comúnmente en áreas tropicales como Indonesia. En este caso, los habitantes del lugar como primeros auxilios recurren a un motor de búsqueda para buscar información sobre enfermedades y medicamentos, especialmente sobre enfermedades generales como la tos, los resfriados y fiebre. No obstante, el resultado del motor de búsqueda convencional genera artículos no relacionados. Además de que no puede manejar el significado sinónimo de la enfermedad en cuestión. De modo que la

información que proporcionan los sitios webs convencionales no es adecuado o preciso para tratar la enfermedad [4]. En este sentido y a este tipo de problemas bien podría dar solución un software de Web Scraping, además de automatizar la búsqueda, brindaría información mucho más relevante y precisa al usuario.

En el campo de la tecnología web, con el paso del tiempo, el desarrollo y la constante transformación de este ha tenido cambios muy importantes, por ejemplo, al enviar correos electrónicos por e-mail o mensajes por aplicaciones de mensajería instantánea, publicar estados en una red social, en un blog, en un repositorio de código, compartir contenidos multimedia, responder una encuesta, se generan a diario una enorme cantidad de información a través de sitios web que no son precisamente estructurados lo que complica su estudio [5]. Es decir que, con la incorporación de las nuevas tecnologías, hay un rápido crecimiento de los usuarios de Internet y de los datos (muchas veces no estructurados) generados por esos usuarios en Internet, afirma [6]. Del mismo modo los sitios corporativos apoyados por el comercio electrónico generan información que pueden servir en la toma de decisiones [7]. De tal manera que los medios digitales se han convertido en un espacio importante para la opinión pública, sea en el ámbito político o comercial.

En base lo expuesto y teniendo en cuenta las herramientas existentes en el mercado que hacen posible el proceso de recolección de datos en ámbitos concretos, además, las herramientas gratuitas o de bajo costo ofrecen funcionalidades muy limitadas las cuales recolectan información, no obstante, los datos que se extraen no facilitan el análisis de datos. De manera que se identifica un problema por lo que no se cuenta con una solución tecnológica que tenga la capacidad de adaptarse a diversos ámbitos, que ofrezca las funcionalidades necesarias de extracción y almacenamiento ágil de datos. Además, es válido indicar que es necesario una herramienta que se adapte a un ámbito de páginas web que permita obtener información proveniente desde las APIs de repositorios de códigos sobre el mantenimiento y evolución del software. ¿A través de qué herramienta se puede recoger, procesar y presentar información proveniente desde las APIs de repositorios de códigos de una manera rápida y sencilla sobre los proyectos, el mantenimiento y/o evolución del software?

Justificación

El manejo de la información se ha vuelto muy importante para las compañías actualmente involucrando así el uso de web scraping para obtener información de forma automática a través de los sitios web, así como redes sociales, blogs, repositorios de código, entre otros. Situación que en su mayoría beneficia a las instituciones o empresas a nivel mundial.

El desarrollo y la constante transformación de la tecnología web ha aportado al crecimiento de las compañías a escala mundial con ello también se han ido generando grandes cantidades de información en la web. Asimismo, el avance de la tecnología web ha beneficiado a los desarrolladores brindándoles información sobre determinados proyectos informáticos. En este contexto, ya sea un curso, o un proyecto de desarrollo de software, también han ido creciendo conforme ha ido pasando el tiempo, por ejemplo, las que se encuentran alojados en un repositorio de código como GitHub disponen información relevante para un desarrollador. En base lo anterior, se puede aplicar web scraping para extraer información sobre diferentes proyectos desarrollados o en desarrollo. De modo que tanto desarrolladores como las empresas tienen a su disposición una gran cantidad de información con lo cual pueden realizar tratamiento y estudiar para mejorar su conocimiento.

La extracción de la información por medio de las páginas web o desde la api de repositorios de código es una tarea muy complicada; si la realizara de forma manual, teniendo en cuenta que cada uno de los mencionados alojan en el Internet millares de información a través de la web que no son precisamente, valiosos para el desarrollador o para las compañías, de modo que es indispensable un programa que provea herramientas necesarias que faciliten la extracción de la información, automatizando el proceso que reducirían el tiempo y costo.

En ese marco, la investigación tiene mucho beneficio para la comunidad desarrolladora, ya que les permitirá extraer información de forma automatizada y estar al tanto de las novedades sobre el desarrollo, actualización de códigos y de nuevos proyectos informáticos alojados en la plataforma de repositorio de códigos. De manera particular, las instituciones o empresas facilitando el manejo de la información para fortificar los procedimientos con respecto a la toma de decisiones, permitiendo la escalabilidad.

El desarrollador conoce estas necesidades de las instituciones, empresas, ciencia de datos, la comunidad educativa, y, sumando a esto que, en la actualidad, la toma de decisión está dada bajo el análisis de datos para esto existen muchas técnicas y herramientas. Por lo que, en base a lo expuesto, se va a generar una investigación sobre web scraping, una de las distintas técnicas de extraer información de sitios web y finalmente desarrollar una aplicación utilizando una de estas técnicas de web scraping.

Objetivos

Objetivo General

Desarrollar una aplicación para extraer información desde la API de un repositorio de código utilizando la técnica de web scraping.

Objetivos Específicos

- Investigar las distintas técnicas y herramientas de extraer información desde las APIS de repositorios de código.
- Realizar un análisis comparativo de estas técnicas que van a ser utilizadas para el desarrollo del software.
- Construir un software utilizando técnicas de web scraping y herramientas actuales de desarrollo.

CAPÍTULO I

MARCO DE REFERENCIA

ANTECEDENTES

Los antecedentes de la investigación desarrollada radican en la similitud de algunos proyectos que se encuentran en distintos repositorios, bibliotecas, revistas y plataformas del Internet.

A continuación, se describe el tema "Implementación de algoritmo en el Lenguaje R para extraer los datos de los Perfiles en Google Scholar utilizando la técnica web Scraping de Minería de datos", el estudio propone determinar la viabilidad sobre la obtención de la información aplicando la técnica en estudio, para lo cual, se ha desarrollado un algoritmo en el lenguaje R que permita obtener datos de forma automática a la vez que permita estructurar los datos obtenidos, disminuyendo así el tiempo de scraping. Para la prueba del algoritmo, se involucraron a las 15 Instituciones de educación superior con distintos tipos de perfiles y publicaciones. Las cuales lógicamente debieron estar registrado en Google Académico. De acuerdo con el estudio, la minería de texto es una herramienta que facilita la obtención de datos estructurado, caso contrario la información debe ser tratada y transformada en información útil para el usuario, de ahí que la técnica de Web Scraping es una alternativa para emplear a la hora de extraer datos de algún sitio web. A pesar de la complejidad que acarrea el desarrollo de esta técnica debido al conocimiento que debe poseer sobre el manejo del lenguaje usado, el estudio considera que implementar un algoritmo, resulta ser ventajoso pues la obtención de la información ya es estructurada y personalizada, además de reducir el tiempo en comparación con otros métodos. Por lo que, con el algoritmo desarrollado en R, se ha optimizado el tiempo de extracción, y con todo esto indica que los resultados del estudio son de mucha utilidad para las instituciones y las personas interesadas, ya que cuentan con un medio para les permite disminuir el tiempo y trabajo de extracción. de la misma manera les facilita el análisis de la información de manera más rápida y oportuna [8].

Por otra parte, se explica en resumen lo que refiere al tema principal aquello que dice la tesis de grado titulada "Desarrollo de un sistema de seguimiento para Instagram", cuyo autor [9], acorde el título, implementó un sistema que permita rastrear las

publicaciones que realizan los distintos usuarios de Instagram y con ello hacer estudios respectivos de los datos recabados. El estudio, además indica que el seguimiento se realizó por usuario o por etiqueta, asimismo incluyó un análisis en base a las opiniones de los internautas y las intercomunicaciones entre ellos. Una vez concluido el desarrollo y probado el sistema, el estudio determina que sí funciona según lo propuesto, pues permite obtener todos los posts de un usuario. Cabe mencionar que la cuenta o el perfil del usuario objeto debe ser público. Con ello el sistema puede extraer hasta 6 mil posts de las últimas publicaciones que hayan realizado, no obstante, las publicaciones deben cumplir con una condición, es que el post debe contener etiqueta. Adicionalmente, enfatiza que es importante tener conocimiento previo, es decir, conocer el nombre exacto de un usuario con fin el de obtener todos sus posts. En cuanto al lenguaje de programación utilizado para el desarrollo del sistema, menciona que ha sido posible gracias al lenguaje Python, junto con ella librerías disponibles o desarrolladas para Instagram. Para el almacenamiento de los datos ha utilizado base de datos MongoDB. Durante el transcurso de la investigación se encontró con algunos imprevistos, ya que Instagram suspendió la función de un endpoint de la Web API, lo que provocó que el estudio buscara otras alternativas y así concluir de manera satisfactoria el trabajo.

Como Proyecto Final de Máster, denominado: "Desarrollo de un buscador de recetas basado en web scraping", se trata de un proyecto viable basado en web, en el que utiliza la técnica de web scraping que permite realizar búsqueda entre un conjunto de páginas web, la misma sea capaz de categorizar o depurar las búsquedas bajo unas reglas específicas en Cook Books. Como, por ejemplo, buscar según los ingredientes o según el tipo de receta. Una vez finalizado el proyecto, la investigación concluye que la utilización marcos de desarrollo, así como para backend y frontend reduce el tiempo, además de obtener productos de calidad, sumándole a esto los resultados de aprendizaje, crecen de forma exponencial. Destaca que el uso de Scrapy como framework para web scraping, Angular como marco de desarrollo para frontend y Slim para el desarrollo de API REST es una opción que se debe tomar en cuenta, pues reduce el tiempo, por ende, también reduce el costo de proyectos. Bajo este estudio, se sabe que el web scraping es una tecnología en auge en la actualidad, en donde las empresas aprovechan para realizar estudios de mercado, mejorar servicios o productos, o para estudiar cualquier tipo de proyectos de desarrollo de software [10].

En el tema: “Desarrollo de una API para datos abiertos”, el estudio hace hincapié en los términos que engloban a los datos abiertos proporcionados por las administraciones públicas y de su situación actual y futura, así como entender el concepto e influencias de la creación de APIS. Para el desarrollo del sistema ha empleado como herramienta principal el lenguaje de programación Python, además de utilizar algunas librerías que le facilitaron la culminación del estudio. Entre sus conclusiones determina que las ramificaciones que existen en las APIs (fintech, insurtech, healthtech) son muy instructivos y atrayentes, por lo cual considera que son temas que en los últimos años han aumentado su relevancia en las empresas y que se van a seguir hablando de ellos, que van a seguir cambiando los modelos tradicionales del negocio. En el estudio ha aplicado herramientas como Django y concretamente Django REST Framework, siendo este último un potente framework de creación de API REST basado en Python 3 [11].

Con un enfoque en el análisis de los datos y orientado a la minería de texto para determinar los temas de los textos, los sentimientos y emociones que estos expresan. Se desarrolla un sistema orientado al análisis descriptivo y predictivo. Por ello, los datos recolectados provienen de fuentes web como sitios de noticias, blog y redes sociales [12], a través de su trabajo de titulación: “Minería de Texto de la Web, de Opinión Pública y Hechos Referentes al Barrio la Floresta”, desarrolla un sistema que tiene la capacidad de clasificar texto según la temática a la que pertenece e identifica los sentimientos y las emociones expresadas en él. El estudio enfatiza que, a pesar de no corresponder de manera explícita a un caso de analítica de Big Data, se basó en una mitología pensada en el análisis de datos no estructurados y de grandes volúmenes y determina que las frases aplicadas se adaptaron perfectamente a las necesidades del proyecto. No obstante, de acuerdo el estudio, un problema inevitable para todo sistema automático de recolección de datos web, es el mantenimiento de este. Esto en el sentido en que las fuentes de datos son específicamente sitios web, los cuales están en constante cambio. Por ejemplo, la renovación de la estructura de un sitio web, al que el sistema accede a recoger datos mediante web scraping, dejaría obsoleta la adquisición de datos de dicha fuente. Y no solamente ocurre con web scraping, sino que sucede a muchas de las herramientas disponibles en la Internet.

BASES TEÓRICAS

Web

(World Wide Web también conocido simplemente como www), es la agrupación de varios documentos (páginas webs) vinculados entre sí a través de links de hipertexto. Desde el punto de vista de [13], sostiene que el "hipertexto" es la combinación de escrituras, ilustraciones y ficheros de cualquier tipo, todo estos en una misma página web. En base lo anterior, se entiende que tanto e-mail, juegos, redes sociales como Facebook, Twitter, así como blogs, wikis y lo demás son parte de la Internet, pero no la web en sí. No obstante, existen dos términos que a menudo se utilizan como es el sitio web y página web. El sitio web es definido como el conjunto de páginas web relacionados entre sí, las cuales son identificadas a través de un nombre conocido como el dominio, el cual normalmente es alojado en un servidor HTTP. De ahí que puede ser accedido por medio de un protocolo IP al utilizar un hipervínculo que permite reconocer el sitio web. Por otro lado, una página web, es un documento que por lo general contiene textos, audios, programas, vídeos, etc. A un sitio web se puede ingresar, básicamente utilizando un navegador web a través de su enlace, en este sentido no es necesario acceder hasta el servidor de alojamiento del sitio web objetivo [14]. Las páginas webs se componen de "elementos", que son unidades con significado, como: títulos, párrafos, lista, tablas, imágenes, etc. Generalmente, los elementos se presentan por medio de etiquetas, estas etiquetas pueden ser de inicio (apertura) y de final (cierre), dentro de ellas está el contenido del documento, que puede ser texto o bien, más elementos. Las etiquetas tienen un nombre y puede abarcar propiedades como las que se puede apreciar en el siguiente ejemplo:

```
1 <h1> Título del artículo </h1>
2 <p> Párrafo </p>
3 <a href="http://tusitio.com">Visita nuestro sitio</a>
4 <img src="casa.jpg">
```

La línea número 1 se refiere a un título. La etiqueta h1 se usa para los títulos principales. Los títulos inferiores se pueden representar con las etiquetas h2, h3 y así sucesivamente. La línea 2 muestra la etiqueta p que usa para describir párrafos de texto. La línea 3 es un enlace, el cual se representa con la etiqueta a. Dentro de este, en el atributo href está

indicada la dirección del enlace. Y la línea 4 es la etiqueta img, con la que se representan imágenes. En el atributo src se coloca la dirección del archivo de la imagen [15].

Protocolo de la web

Para que todas estas técnicas funcionen como cualquier aplicación basados en web debe regirse en ciertas reglas, como es el protocolo de la web. Por tanto, un sitio web para poder brindar o ejecutar las peticiones solicitadas debe usar determinadas tecnologías. Esta acción es conocido como la interoperabilidad entre distintas aplicaciones. Aquí es donde también intervienen ciertos protocolos y normas que determinan el modo de comunicación, la forma de los datos que son enviados y recibidos, al igual que su funcionamiento, entre otros. A continuación, se detallan los protocolos y normas fundamentales para que un servicio web pueda operar y ser utilizado [1]:

- Uno de los formatos es el XML (es la sigla de Xtensible Markup Language), en otras palabras, es el lenguaje de marcado que facilita la escritura de los contenidos permitiéndoles dividir de su formato. Cabe mencionar que esta norma contiene a su vez las normas como el DTD o XSD, la cuales permiten la configuración del lenguaje y el XSL-FO y XSLT las cuales sirven para la conversión y presentación de la información [16].
- El formato JSON (es sigla de JavaScript Object Notation), es un tipo de lenguaje que permite guardar e intercambiar información, tiene un formato sencillo, de tal manera que puede ser leído y escrito por personas [16].
- El estándar SOAP (es la sigla de la siguiente: Simple Object Access Protocol), esta norma es considerado como el más difícil, ya que funciona bajo una cascada de marcado que XM, al igual que el anterior, este también es utilizado para intercambiar información, para lo cual se cumple lo siguiente: una parte que se realiza la petición (cliente) y una parte que responde dicha petición (servicio), básicamente esta norma se utiliza en el ámbito de los servicios web [17].
- REST (esta sigla corresponde a las siguientes: Representational State Transfer), es una norma de envío de representación de estado, hay que tener en cuenta que este no posee estado (del inglés stateless), lo que significa que, cuando exista dos requerimientos sea cual sea, el servicio tiende a perder toda su información [18].

- El protocolo WSDL (sigla que representa a las siguientes: Web Services Description Language), este protocolo o norma se basa en el formato XML, sólo para la interfaz del servicio web, es decir, los métodos y criterios que se muestran así como la entrada y salida para llamar a los servicios [17].

Web Scraping

Web Scraping (traducido al español sería raspado de páginas web), es una práctica que va tomando fuerza a medida que transcurre el tiempo dentro de las empresas o incluso dentro de las instituciones educativas. En términos simples, es una forma de obtener datos de una o varias páginas web de forma automática, esto incluye tales como redes sociales, repositorios de código, blog, tiendas online, sitios empresariales, entre otros, aplicando algún método que actualmente presenta esta técnica [8], [19]. Es considerado como una técnica de programación ya que facilita la extracción datos de la World Wide Web, es decir, páginas web. En algunos casos, estos tipos de software van mucho más de allá de una simple programación, dicho de otra manera, los programas son dotados de inteligencia artificial, lo que genera más autonomía posibilitando una navegación continua por Internet y extraer información relevante. Para el desarrollo de esta técnica se puede hacer mediante una variedad de lenguajes de programación que admiten la programación del protocolo de Transferencia de Hipertexto [20]. Como lo afirman [21], [22], el objetivo principal de Web Scraping, es conseguir cantidades enormes de información, mediante algoritmos de búsqueda de los cuales pueden rastrear centenares de sitios web, esta actividad se lleva a cabo, normalmente en páginas que utilizan lenguajes de marcado como HTML o XHTML, de tal manera que, es necesario conocer cómo está organizada la información de la página web de la cual se desea extraer información. Bajo este criterio se puede asumir que el web scraping es la solución intermedia entre la recolección manual de datos (marcando, copiando y pegando textos) y el acceso automatizado a los mismos con base en un protocolo predeterminado (API, framework, librerías, etc.). Básicamente, esta práctica se aplica cuando tales protocolos no existen y la cantidad de datos que se desea extraer es demasiado grande para que pueda ser realizada en forma manual [15]. El Web Scraping también ayuda a la automatización de la web de muchas maneras, incluyendo datos meteorológicos, detección de cambios en la web y comparación de precios de sitios web en línea, entre otros. Esta técnica, puede convertir información no estructurado en información estructurado, luego de ello, una vez

que sea validado, se procede a guardarlos en una base de datos [23]. Posteriormente, se pueden analizar y conocer el código de HTML devuelto por cualquier sitios tras una petición HTTP:GET [24].

El término "scraping" implica que la extracción de la información puede ser de cualquier fuente como base de datos, archivos CSV, repositorios de códigos como Github, Bitbucket o GitLab, por lo que no necesariamente la fuente de extracción de información debe ser netamente Sitios Web Comerciales o tiendas online.

Crawler

Otro término que se relaciona mucho con el web scraping o raspado web, es el web crawling, web crawler o simplemente crawler (rastreador web). Un método que tiene como fin obtener los hipervínculos, es decir, los enlaces o links. Una vez hecho aquello, el programa inicia visitando los links de las páginas, y va encontrando más y más links hasta encontrar la información requerida [25].

Tipos de Web Scraping

De acuerdo con Leonardo Kuffo, del sitio web udemy.com, existen varias formas de realizar web scraping, no obstante, existen las dos más importantes, de las cuales se tratará a continuación:

- **Web Scraping de una sola página web**

  Tipo de web scraping que se realiza a una sola página de un sitio web. Para este proceso, la librería Scrapy dispone de una clase s*crapy.spiders.Spider*.

- **Web Scraping de varias páginas web**

  Como lo hace notar su nombre, esta clasificación realiza web scraping a varias páginas de un sitio web. Para ello, la librería Scrapy también cuenta con una clase *scrapy.spider.CrawlSpider*.

Cuando se realiza web scraping a varias páginas web intervienen dos técnicas: Crawler vertical y Crawler horizontal.

Crawler vertical: cumple la tarea de extraer los ítems especificados de una página web (GitHub en este caso), en las siguientes imágenes (Figura *1* y Figura *2*) se observa los repositorios e ítems de este.

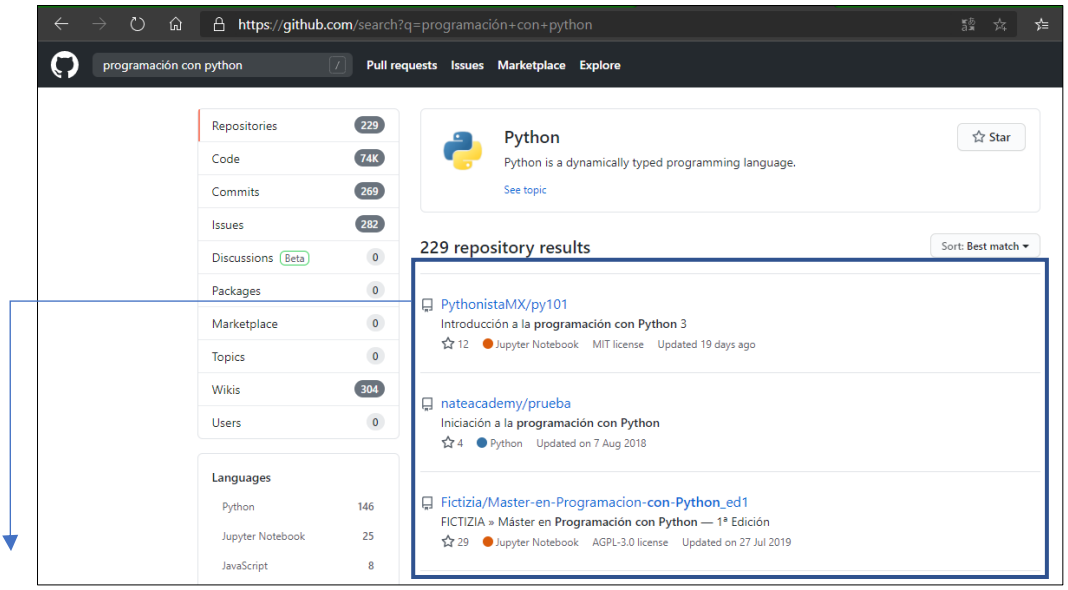


Figura 1. Crawler vertical.
Fuente: Plataforma Web GitHub (1/2)

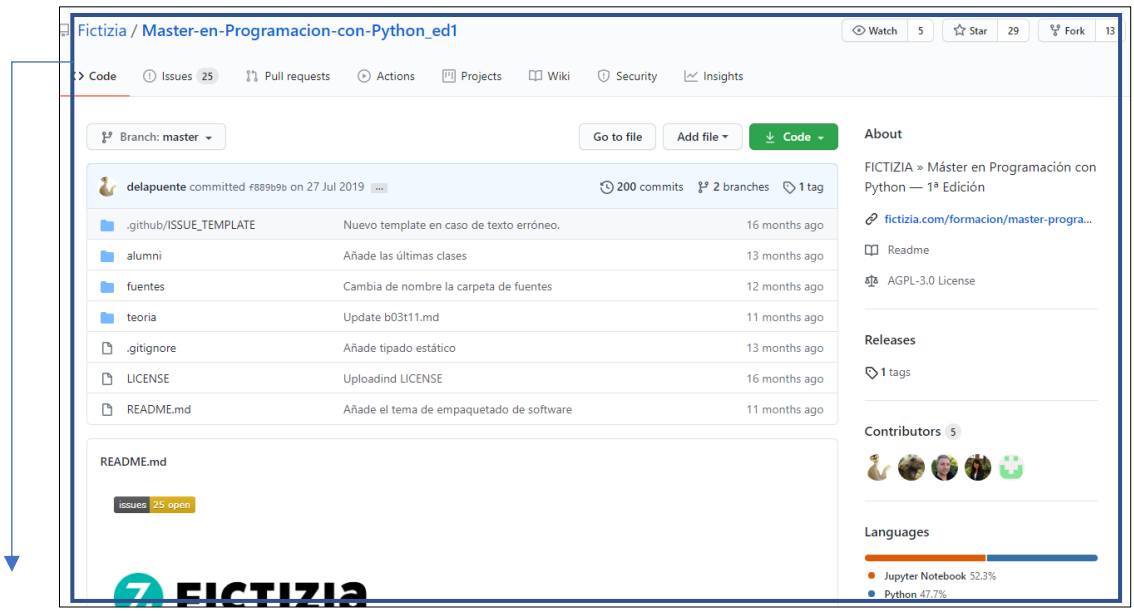


Figura 2. Crawler vertical.
Fuente: Plataforma Web GitHub (2/2)

Crawler horizontal: mientras que este cumple la función de visitar las paginas (número de páginas) del sitio web según se haya especificado en la petición. Básicamente, las dos

técnicas van de la mano para realizar web scraping de varias páginas de un sitio web (GitHub en este caso), como se observa en la siguiente ilustración (Figura *3*).

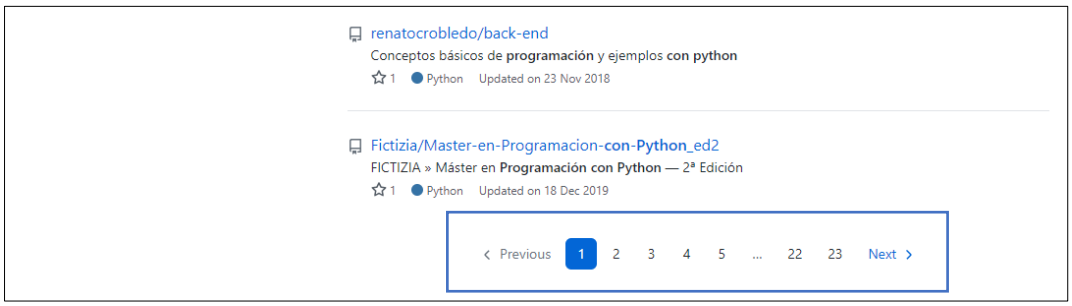


Figura 3. Crawler horizontal (Paginación)

Fuente: Página Web GitHub

Web Scraping con librerías

El proceso de Web Scraping se compone de estos sencillos pasos fundamentales:

1. URL semilla

   El primer paso para tener siempre a consideración es tener una url semilla, pues se parte desde aquí.

   Básicamente una url semilla, es aquel sitio web del cual se hará web scraping.
2. Request

   A través del método request se realiza requerimientos a la url semilla, es decir, se le indica qué información o qué partes de url se quiere extraer información.
3. Response

   Luego de ello se obtiene una respuesta de la url, esta respuesta puede ser en formato XML o HTML, que posteriormente será parseado, es decir, se obtendrán los ítems especificados en el requests.
4. Populate Items

   Los ítems dependen de la página web, según su estructura. Desde ellos se obtiene la información deseada.
5. More URLs

   A partir de la URL semilla se puede ir a más URLs obteniendo así la información deseada. Y de estas urls se repite los pasos indicados anteriormente.

Ventajas

- No se depende de un API: La ventaja más importante que tiene Web Scraping es la independencia del API, por lo cual no existe límite de ningún tipo.
- No tiene limitaciones: No tiene límite en el tiempo de extraer información (rate limit), así como tampoco tiene límite sobre qué información se desea extraer.

Desventajas

- Siempre dependerá de la estructura de XML o HTML de la página a la cual se quiere realizar scraping.
- Pueden banear la IP: Una web scraping al ser una actividad repetitiva, si no se aplica de manera adecuada esta técnica, puede resultar invasivo para el sitio objetivo o incluso puede ser visto como un ataque al sistema. En este sentido y teniendo en cuenta, además que algunos sitios web disponen de mecanismos de seguridad, pueden bloquear la IP del visitante, inhabilitando así la posibilidad de realizar más visitas o peticiones.

La técnica de Web Scraping requiere de dos actores importantes para llevar a cabo, estos son:

HTML

La sigla HTML corresponde a los siguientes términos HyperText Markup Language (en español conocido como Lenguaje Marcado de Hipertexto). De acuerdo con [26], es el lenguaje de marcado más utilizado para desarrollar aplicaciones web. "Hipertexto" hace referencia a los hipervínculos que permiten interconectar las diferentes páginas web, de hecho, esto puede existir dentro del mismo o a través de distintos sitios web. El término "marcado" hace referencia a aquello que permite cargar ya sea textos, imágenes o algún otro contenido que se quiera incluir en una página web con el objetivo de mostrar por medio del navegador web. HTML abarca todo lo que son los elementos especiales, por ejemplo, <head>, <title>, <body>, <header>, <section>, <p>, entre otros [27].

HTTP

HTTP es una sigla que representa a los siguientes términos HyperText Transfer Protocol, en español denominado Protocolo de Transferencia de Hipertextos. Es una regla para la capa de aplicación que sirve como canal por el cual son transferidos un conjunto de hipertextos y multimedios conocido como hipermedia, esto es el propio HTML, o bien es el protocolo de transmisión de información de la Word Wide Web. Fue creado para la interacción entre los diferentes servidores y navegadores web. De ahí que surge el paradigma cliente-servidor, donde el cliente (usuario) se conecta con el fin de enviar una solicitud hacia el servidor, por la cual recibe una respuesta de parte de este [28], [29].

Distintas herramientas de extracción de información de páginas web

Las técnicas o herramientas que existen para implementar una aplicación que permita realizar web scraping son muy variados, las cuales se pueden considerar en tres categorías principales: bibliotecas para lenguajes de programación de uso general, frameworks y entornos de escritorio.

Librerías

Una de las técnicas más común o preferido por los desarrolladores consiste en construir sus propios webs scraping utilizando algún lenguaje de programación según su preferencia. Para aquello utilizan las bibliotecas de terceros, las cuales permiten el acceso al sitio implementando el lado del cliente del protocolo HTTP. Por lo cual, a continuación, se detallan algunas de las bibliotecas más populares que permiten el acceso a los sitios web:

libcurl

(http://curl.haxx.se/). [30], soporta las principales características del protocolo HTTP, incluyendo certificados SSL, HTTP POST, HTTP PUT, FTPuploading, carga basada en formularios HTTP, proxies, cookies y autenticación HTTP.

Beautiful Soup

Es otra de las librerías que se encuentra disponible para la técnica de web scraping. Una librería para Python que cuenta con todas las herramientas necesarias, otro dato relevante de esta herramienta es que, es gratis. Es totalmente configurable y personalizable. En el caso de que no tenga conocimiento sobre Python la curva de aprendizaje crece de forma exponencial, ya que el usuario debe configurar todo, esto implica que la configuración sea solo a nivel de sistemas locales ya que, si se instala en arquitecturas cloud, se pierde el uso gratuito de la biblioteca [31].

Requests

Es una biblioteca HTTP de Python, lanzada bajo la licencia Apache 2.0. su objetivo es facilitar las solicitudes HTTP haciendo más simples para los usuarios [32], [33].

Jsoup

Es una librería de Java. Se utiliza junto con HTML, con la ayuda de la API que esta librería posee. Una API que sirve para obtener y depurar datos, mediante el formato DOM y CSS. Cumple las funciones como: raspar y analizar datos HTML (desde un enlace, fichero o una serie de textos), buscar y extraer información (utilizando el modelo en objeto o los selectores del formato CSS), así como también la manipulación de los componentes de HTML (es decir, las características y los textos), también cumple la función de depurador (los contenidos enviados por el usuario), finalmente procurar que la salida de hipertextos sea lo más organizada posible [34].

Framework

Scrapy

(http://scrapy.org) es un marco de desarrollo enfocada a las aplicaciones que realizan el rastreo de sitios web y extraen información. Es utilizado y aplicado en distintos proyectos orientadas al web scraping, principalmente creado para el lenguaje Python. Con este framework las páginas web se analizan automáticamente y los contenidos web se

extraen usando expresiones XPath [35]. En la siguiente ilustración (Figura 4) se puede observar la arquitectura de Scrapy al igual que sus elementos y la secuencia de datos que son parte de la librería. Así mismo, mediante una tabla (Tabla 1), de forma resumida se realiza descripciones de cada uno de los elementos.

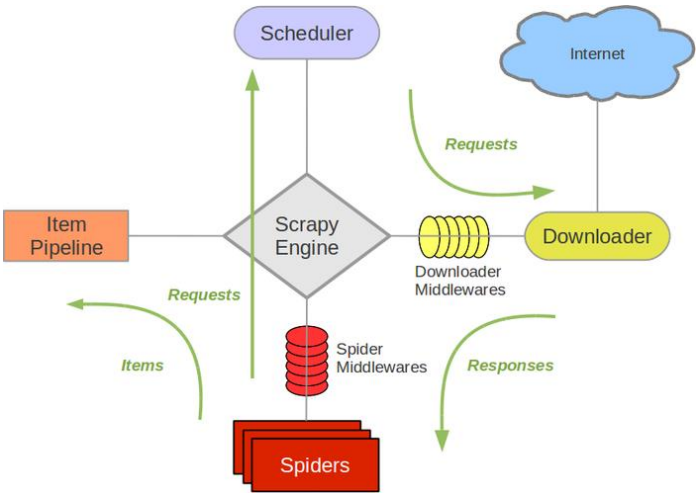


Figura 4: Arquitectura y componentes de Scrapy [36]

Tabla 1. Descripción de los componentes de Scrapy [10]

Componente	Descripción
Scrapy Engine Motor de Scrapy	Es el componente que se encarga de monitorizar la secuencia de datos dentro del sistema y es capaz de lanzar eventos en caso de que se realiza cierta acción.
Schedule Planificador	Este componente es el responsable de receptar peticiones del motor, una vez recibido las coloca en cola para luego devolver al motor para cuando éste las necesita.
Downloader Descargador	Este elemento es el encargado de realizar descarga de las páginas web y suministrar al motor y que este a su vez, suministra a las arañas web.
Spiders Arañas	Este componente es personalizable, es decir, en base un análisis o según lo que se requiera los ítems pueden ser extraídos de las páginas web scrapeadas.
Item Pipeline Tubería o canal de elementos	El canal o la tubería de ítems, es la que se encarga del tratamiento de los datos, después de que han sido sacados por los spiders. Las actividades típicas de este elemento es la depuración, verificación o la persistencia, dicho de otra forma, guardarlos en una base de datos.

Web-Harvest

(http://web-harvest.sourceforge.net/), es un framework diseñado para un lenguaje de programación específicos de un dominio (DSL). Por ejemplo, en este caso está diseñado para el lenguaje Java. Los procesos de extracción web se describen en XML [37].

Tabla 2. Librerías y frameworks de web scraping de código abierto [38]

	Tipo C: HTTP cliente P: Parsing F: Framework	**Idioma de dominio específico**	API/ independiente	**Lenguaje**	**Facilidades de extracción** R: expresión regular H: Árbol de análisis HTML X: XPath C: CSS selectors
UNIX shell (curl, wget, grep, sed, cut, paste, awk)	CP	No	SA	Bash	R
Curl/libcurl	C	No	Both	C+ bindings	
Web-Harvest	F	Yes	Both	Java	RX
Jsoup	CP	No	API	Java	HC
httpClient	C	No	API	Java	
jARVEST	F	Yes	Both	jRuby/Java	RXC
WWW:: Mechanize	CP	No	API	Perl	RX
Scrapy	F	No	Both	Python	RX
BeautifulSoup	P	No	No	Python	H
Requests	P		No	Python	H

La Tabla *2*, es un resumen comparativo de las bibliotecas y frameworks más populares de código abierto para web scraping, en base la cual se puede indicar que los lenguajes, que más cuentan con herramientas para el desarrollo de web scraping son Java y Python. Sin embargo, en los apartados posteriores se estudiará más lenguajes que son potencialmente recomendables para este fin.

Web Scraping con las APIs

Este proceso es el más sencillo teniendo en cuenta que es la técnica proporcionada por la misma plataforma para realizar este tipo de trabajos, solamente se debe usar el api correcto según los datos que se requiere obtener. En este sentido la documentación de GitHub proporciona lo necesario para los distintos fines, por ejemplo, existe para buscar los commits de los repositorios, para buscar repositorios de uno mismo o de otros usuarios, asimismo existe api para buscar los determinados lenguajes de programación o cualquier otra herramienta utilizada en el desarrollo de proyectos o desarrollo de aplicaciones. La Tabla *3*, muestra algunos ejemplos del uso de la API de la documentación de GitHub para hacer Web Scraping mediante apis:

Tabla 3. API de GitHub para Web Scraping [39]

Dominio	API
"user_repositories_url":	"https://api.github.com/users/{user}/repos{?type,page,per_page,sort}",
"current_user_repositories_url":	"https://api.github.com/user/repos{?type,page,per_page,sort}",
"commit_search_url":	"https://api.github.com/search/commits?q={query}{&page,per_page,sort,order}",
"followers_url":	"https://api.github.com/user/followers",
"repository_search_url":	"https://api.github.com/search/repositories?q={query}{&page,per_page,sort,order}",
"user_organizations_url":	"https://api.github.com/user/orgs",
"rate_limit_url":	"https://api.github.com/rate_limit",

Fuente: Elaboración propia

La Tabla *3*, muestra una pequeña parte de la documentación de las APIS de GitHub con las cuales se puede realizar web scraping desde los apis, es decir, sin necesidad de utilizar framework como Scrapy.

Entornos de escritorio

Generalmente, un software de web scraping viene integrado un navegador, donde el usuario puede navegar hasta la Web de destino y seleccionar de forma interactiva los elementos de la página a extraer, evitando cualquier especificación de expresiones regulares, consultas XPath u otros tecnicismos. Además, hay módulos disponibles para crear múltiples tipos de salida, como archivos en formato CSV, Excel y XML, e inserciones en bases de datos. Los principales inconvenientes de las soluciones de escritorio son la distribución comercial y el acceso limitado a la API, lo que dificulta la integración de estos scraping en otros programas (lo que a menudo es un requisito) [24]. Las herramientas de web scraping son muy numerosas, a continuación, se citan en lista las herramientas de extracción de datos ordenadas alfabéticamente:

Apify.com

Para poder utilizar esta herramienta, es bastante recomendable tener conocimientos básicos de JavaScript. Esta herramienta permite conectarse a una API y obtener los datos en formatos tipo CSV, JSON, XML, RSS, etc. Ofrece tanto plan gratuito y planes de pago

que parten de los 49 dólares mensuales, lo que de alguna manera la ubica para nada en los puestos de las más caras. De acuerdo con su portal web, tienen dos opciones principales. Primera, más pensada para desarrolladores para aquellos que pretenden obtener datos de las webs en bruto, y la segunda, más enfocada a capas de negocio, en el que lo importante es la funcionalidad con la que cuenta la herramienta, es decir la conceptualización y análisis de la información, con lo cual puede proceder a una toma de decisiones adecuada. Es decir, un nivel de consultoría de datos propiamente dicho. Además, disponen de documentación de calidad en la web y de ejemplos de casos de estudio. No dispone de una interfaz gráfica que sea bastante utilizable, debido a que está más enfocado para programadores, ya que su servicio es una librería utilizable a nivel de API [40].

Dexi.io

No dispone de un plan gratuito de prueba. Permite utilizar Arañas, y bots, entre otros sistemas. Sin embargo, su curva de aprendizaje es algo más elevada a diferencia de otras herramientas. Por lo que es considerado como "la herramienta de web scraping para usuarios avanzados". No obstante, a pesar de su alta complejidad, sí dispone de ciertos apoyos basados en herramientas visuales que permiten organizar la arquitectura de un crawler a nivel visual. El gran potencial de esta herramienta es que sus bots no se aplican solo a la extracción recursiva y pura de datos, sino que pueden ser automatizados para realizar tareas sencillas y profundizar mejor en los datos deseados [41].

Diffbot.com

El mayor aporte de valor de esta herramienta consiste en la utilización de Inteligencia Artificial para mejorar la capacidad de filtración y extracción de datos. Prueba de 14 días gratuita, pero, después, los precios más baratos parten de los 300 dólares. Su potencial radica en que, con ayuda de la inteligencia artificial, puede mejorar la búsqueda de elementos sin estar limitados a simples búsquedas por estilos o querys de elementos [42].

Hunter.io

Se puede considerar como una herramienta de extracción de datos, pero centrada en correos electrónicos. Por ende, sale de la temática de "extracción de datos en páginas web". Se trata, principalmente, de un buscador de emails en dominios personalizados. Útil en caso de que se requiera contactar con alguna compañía. La búsqueda puede ser en ambos sentidos. Es decir: se puede buscar correos de un dominio concreto (@dominio.com) o también se puede buscar un nombre de un usuario y ver en qué dominios está adscrito un correo electrónico bajo ese alias, aunque esta segunda no funciona del todo bien. De acuerdo con la web de la herramienta son usuarios las empresas como Google, IBM, Manpower, Microsoft, Adobe e Invision [43].

Import.io

Una de las más utilizadas para ejemplificar el concepto de web scraping. Muy recomendada desde sitios como ecomaster para analizar a la competencia de un entorno e-commerce. Cuenta con una interfaz de usuario amigable, sin embargo, su mayor inconveniente y la que más limita su número de usuarios, es que es muy cara: entre 300 dólares mensuales o 2 000 anuales para los planes básicos. Posee versión gratuita, pero solo dura 48 horas, lo que la convierte en mala candidata para realizar investigaciones y pruebas de concepto [44].

Mozenda.com

Definida como la mejor opción para las empresas que buscan una plataforma basada en la nube. Tienen almacenada gran cantidad de información de todas las páginas que han *escrapeado*, por lo que se permiten posicionarse, además, como una empresa DAAS (Data As A Service). De manera que, su negocio no va enfocado exclusivamente a ofrecer una herramienta útil de extracción de datos, sino que es una de sus ramas. Debido a que se debe pagar por una licencia mínima de 250 dólares, queda descartada para pruebas de concepto [45].

Octoparse

El equipo de ScraperApi los definen como una buena herramienta para personas que buscan extraer datos de sitios web sin tener que codificar nada. Los planes de pago no son los más baratos, pero tampoco son excesivos. Parten desde los 75 dólares, pero ojo al dato, tiene un plan gratuito ilimitado muy generoso. Lo que buscan ofrecer, ante todo, es buena experiencia de usuario. Como expresan en su propia web *"point, click and extract"*. No permite configurar muy a bajo nivel el algoritmo del crawler, así que, seguramente, funcionará mejor en unos sitios que en otros. Pero, sin duda, con ese extra de usabilidad que ofrece, tienen las puertas abiertas a gran cantidad de público [46].

ParseHub

Es una buena opción para crear crawlers no requiere programación en ningún lenguaje. Puede ser utilizado por alguien con altos conocimientos técnicos (como un Data Scientist) o por perfiles menos profundizados como periodistas o analistas de datos. De tal forma que, permite que el usuario final se centre en lo importante: tratar los datos. Sin perder excesivos recursos en su obtención. Además, hablan bien de su plan gratuito, lo que lo convierte en muy buena opción para probar en un entorno de investigación y aprendizaje como este. Toda la personalización del crawler puede llevarse de forma visual. Dotándolo de mayor potencia y usabilidad, pero también lastrando la simpleza [47].

ScraperApi

Clasificada como la herramienta para desarrolladores. Permite crear scrapers para analizar la WEB, gestionar proxies e incluso CAPTCHA. Utiliza una API, de manera que su uso resultante bastante sencillo. Eso sí no es un servicio de datafiniti, en otras palabras, no ofrecen datos ya parseados de forma estructurada a través de una API, tan solo utiliza la API para controlar los crawlers. Es una herramienta muy popular por lo que posee un buen estatus entre la comunidad de Open Source, ya que colabora bastante. Su documentación es muy buena y está disponible para quienes quieran echar un vistazo. Permite iniciar sesión automáticamente con los servicios de Google y con las credenciales

de GitHub. Los precios son bastante competitivos, partiendo desde los 29 dólares. Además, se puede hacer las primeras mil llamadas a la API de forma gratuita [48].

Webhose.io

Para hacer uso de esta herramienta no es necesario que uno mismo escrapea diversos sitios, ya que la misma herramienta tiene sus propios scrapeadores de datos y un servicio de estructuración, por lo que los datos extraídos desde sitios ya vienen estructurados, esto es posible al enlace que se realiza a una API, a la que se puede efectuar 1 000 peticiones al mes de forma gratuita. Los datos obtenidos, ya vienen en formatos estandarizados y estructurados como XML o JSON. Facilidad que brinda para obtener los datos deseados de manera estructurada a través de una API hace que los perfiles más técnicos tiendan hacia este a simple vista. Sin embargo, su punto fuerte puede ser también su tendón de Aquiles: ya que es necesario contar con conocimientos medios / altos en diversos lenguajes y tecnologías para poder hacer uso de su API. El costo varía según la cantidad de peticiones que se hace a su API. Permite realizar 1 000 peticiones mensualmente de forma gratuita, en cambio sí se requiere realizar un millón de extracciones, el costo ronda entre los $ 4 000 [49].

80legs.com

De acuerdo con los creadores, es una herramienta de "scraping fácil", esta herramienta básicamente ofrece dos servicios que son fundamentales en el ámbito de raspados web:

Crawling customizado: con esta funcionalidad es posible configurar escrapeadores propios y extraer datos según sea necesario.

Datafiniti: "Omite el scraping y obtén acceso instantáneo a los datos de la web", así es como se definen los desarrolladores, como las demás, ofrecen capacidad de acceder a un api para tener acceso instantáneo a datos sin perder el tiempo parseando sitio, ya que ellos mismos ya se encargan de aquellos. Ofrecen un gran plan gratuito, la misma que no son precisamente los más caros del mercado. Adicionalmente, ofrecen servicios (facturados aparte) en los que son ellos los que crean el crawler a medida si el usuario no

sabe hacerlo y busca datos específicos que no están todavía en su otra empresa, Datafiniti [50].

Tabla 4: Matriz general de clasificación [24]

NOMBRE / WEB	CRAWLING PERSONALIZADO	INTERFAZ GRÁFICA	ML / IA	PLAN GRATUITO	DATAFINITI / DAAS	REPOSITORIOS PÚBLICOS	DOCUMENTACIÓN DE CALIDAD	PLANES DE PAGO	
								MÁS BARATO	MÁS CARO
Apify.com	SÍ	NO	NO	SÍ	NO	SÍ	SÍ	$ 49/MES	$ 499/MES
Beautiful Soup	SÍ	NO	NO	SÍ	NO	NO	MALA	0	0
Dexi.io	SÍ	SÍ	NO	NO	NO	NO	SÍ	$ 119/MES	$ 699/MES
Diffbot.com	SÍ	NO	SÍ	LIMITADO	SÍ	NO	SÍ	$ 299/MES	$ 3999/MES
Hunter.io	NO	NO	NO	LIMITADO	NO	NO	SÍ	$ 34/MES	$ 399/MES
Import.io	SI	SÍ	SI	NO	SÍ	NO	SÍ	$ 299/MES	$ 1999++/AÑO
Mozenda.com	SÍ	SÍ	NO	NO	SÍ	NO	SÍ	$ 250/MES	$ 450+/MES
Octoparse	NO	SÍ	NO	SÍ	SÍ	NO	MALA	$ 75/MES	$ 4899/MES
ParseHub	SÍ	SÍ	NO	SI	NO	NO	SÍ	$ 149/MES	$ 499/MES
ScraperApi	SI	NO	NO	LIMITADO	NO	NO	SÍ	$ 29/MES	$ 249/MES
Scrapy	SÍ	NO	NO	SI	NO	NO	SÍ	0	0
Webhose	NO	NO	NO	LIMITADO	SÍ	NO	SÍ	$ 50/2000 req	$ 4000/1M req
80legs.com	SÍ	NO	NO	SÍ	SÍ	NO	SÍ	$ 29/MES	$ 299/MES
Jsoup	SÍ	NO	NO	SÍ	NO	NO	SÍ	0	0
CARACTERÍSTICAS SERVICIOS WEB SCRAPING									

De acuerdo con la Tabla 4, se puede manifestar lo siguiente: la existencia de distintas herramientas de web scraping, muestra la importancia del tema, que realmente web scraping es una tendencia actualmente para la obtención automática de la información. Siendo estas herramientas librerías, software para escritorio o aplicaciones web. La única limitante sobre el uso de estas aplicaciones son los precios, son elevados en algunos casos, sólo por mencionar alguno, por ejemplo, Octoparse tiene un costo de $ 4 899 dólares en su versión completa. En este sentido no se puede dejar de hablar de versiones gratuitas, aunque con menor capacidad y limitada extracción de datos.

Lenguajes de programación

En este espacio se analiza sobre los distintos lenguajes de programación que se emplean en la creación de aplicaciones enfocadas en la técnica de Web Scraping. Existen varios lenguajes que se pueden aplicar en este campo, no obstante, se tomarán en consideración solo algunos lenguajes que actualmente son más usados para este fin como las que se presenta, a continuación:

Java

Es un lenguaje de programación, en otras palabras, es un entorno de desarrollo. Existe una serie de bibliotecas avanzadas para este lenguaje. Es un lenguaje creado por James Gosling en 1991 para la compañía Sun Microsystems Inc. En el año 2010, la plataforma fue adquirido por Oracle, propiedad a la que hoy en día pertenece. Es un lenguaje que ha ganado mucha popularidad debido a que las aplicaciones Java son capaces de correr en cualquier plataforma ya sea que este se encuentre bajo el sistema operativo, por ejemplo, Windows, Mac OS, Linux o Solaris. Java permite generar programas portables, las cuales funcionan a través de Java Runtime Environment o simplemente conocido como JRE. Java no cuenta con un editor propio, no obstante, existen IDEs que permiten desarrollar aplicaciones de forma simple, entre las que se puede mencionar está NetBeans, Eclipse, JBuilder, JDeveloper. Sin embargo, aún con todo esto, se requiere la instalación de JDK (siglas de Java Developmente Kit), la cual es el Kit de Desarrollo de Java [51]–[53]. En la Tabla 2, se puede observar cada de una de las librerías existentes para java con afines al web scraping.

Python

Es un lenguaje de programación que pertenece a la familia de lenguajes interpretados. Desarrollado bajo un enfoque de licencia Open Source avalada por la OSI (siglas que corresponde a los siguientes términos: Open System Interconnection, en español conocido como Interconexión de Sistemas Abiertos), de manera que se puede usar y distribuir de forma libre, hasta para uso comercial. La licencia de Python es administrada por Python Software Foundation [54]. Como afirma [54], Python es considerado como el lenguaje de propósito general, es decir, bien puede ser aplicado en cualquier ámbito de desarrollo, por ejemplo, programas web, gestión de sistemas, ciencias de datos, ciencia computacional, inteligencia artificial (IA), Internet de las cosas (IoT), entre otros. Cuenta con alta gama de librerías estandarizadas y una variedad de bibliotecas desarrolladas por terceros, siendo así que hoy en día existe más de 100 mil módulos para este lenguaje. Es multiparadigma (Programación Orientado a Objetos o solamente POO, funcional, etc.), los recursos educativos son dispuesto de manera que puedan ser accedidos sin ninguna restricción [56]. En la parte de librerías (Tabla 2), se puede observar cada una de las bibliotecas disponibles para este lenguaje relacionados con web scraping.

Existe una herramienta para Python que permite realizar interfaz gráfica. Su configuración es sencilla y fácil de manejo cuya herramienta es TkInter.

C++

Este lenguaje utiliza el paradigma de programación orientado a objetos, surge a partir del lenguaje C por lo que posee casi las mismas estructuras. Existe varios IDEs para este lenguaje como NetBeans, Dev C++, este último es el más ligero por lo que no ocupa mucho espacio en el disco de almacenamiento, lo que hace ideal para crear programas pequeños que no impliquen la demostración del uso de estructura de control y estructuras de datos [57]. De acuerdo con [58], existe más de 3 000 bibliotecas para este lenguaje para diferentes propósitos. Con base lo anterior, [59] considera que el lenguaje C++ es el adecuado para desarrollar aplicaciones para el raspado web, incluso lo pone por encima del lenguaje Python. No obstante, aconseja no usar a las empresas debido a su complejidad en los códigos.

PHP

(acrónimo de Personal Hypertexto Preprocessor). Es un lenguaje de programación Open Source para crear aplicaciones web [60]. Como afirma [59], PHP es ideal para el web scraping, es tanto así que Import.io y Kimono Labs, aplicaciones para web scraping, se basan en este lenguaje. Infortunadamente, PHP no cuenta con el soporte de la librería Beautiful Soup y Scrapy (que se basan en Python) que son las herramientas de extracción de datos. Como expresa [61], existen varios estudios donde se ha aplicado este lenguaje enfocadas en la técnica de web scraping. Este lenguaje requiere de un servidor local, que cuente con soporte, para funcionar [62].

Perl

Larry Wall creó este lenguaje en el año 1987. Un lenguaje de programación que tiene como finalidad facilitar gestión de actividades, la integración de objetos, referencias y módulos; todo ello ha hecho que este lenguaje sea muy utilizado, tanto así que, una de

sus aplicaciones más importante es el desarrollo de herramientas bioinformáticas. Al igual que el lenguaje Python, Perl también es un lenguaje interpretado, técnica que, dadas las instrucciones por el programador pueden ser ejecutadas por el ordenador sin necesidad de leer y traducir exhaustivamente todo el código [63]. De esta manera, Perl también es considerado como una opción muy importante para el desarrollo de raspado web. A juicio de [64], tanto Python como Perl hoy en día son los favoritos para aplicar en el desarrollo del software enfocadas a las técnicas de scraping.

Tabla 5: Matriz comparativa de lenguajes de programación

Lenguajes de programación	**Librería Web Scraping**	**Servidor local**	**Open Source**
Java	X	-	X
Python	X	-	X
C++	X	-	X
PHP	X	Xampp, Wamp	X
Perl	X	-	X

Fuente: Elaboración propia

El cuadro anterior (Tabla *5*), es un resumen de los distintos lenguajes que se pueden usar para desarrollar aplicaciones para Web Scraping. Hacer hincapié en el lenguaje Python, ya que, además, cuenta con una alta gama de librerías estandarizadas y una gran cantidad de módulos de terceros. En este sentido existen bibliotecas especializadas para Python que permiten realizar web scraping, por ejemplo, Beautiful Soup, Scrapy y PyGithub. En el caso de Beautiful Soup su función principal es extraer datos de archivos XML y HTML. Funciona bajo los analizadores sintácticos como LXML o HTML5LIB para proporcionar diversas formas de navegación, búsqueda y modificación del árbol de análisis sintáctico extraído de una página web mediante uso de requests. Requests es una librería escrita en Python para realizar peticiones HTTP [65].

Framework de desarrollo

Django

Django es un marco de desarrollo de alto nivel para crear aplicaciones web en Python. Presenta un diseño amigable y práctico, además, es libre y de código abierto [66].

De acuerdo con el portal Web EspiFreelancer, este framework se puede definir de la siguiente manera:

- Django sigue el patrón de diseño MTV. Las siglas MTV representan a los términos Model (Modelo), Template (Plantilla) y View (Vista). En este sentido, el paradigma MTV es bastante parecido al paradigma MVC, de tal modo que se podría afirmar que Django es un framework o marco de desarrollo MVC [67].
- Modelo Plantilla Vista (Model Template View o MTV)

  Django también tiene al controlador, no obstante, en forma intrínseca; en consecuencia, el framework completo es considerado como el controlador.

  - **Modelo (Model):** es el responsable sobre lo concerniente a los datos, es decir, cómo se debe acceder a una determinación función del sistema, la validación, y todo lo relacionado con la información y su comportamiento.
  - **Plantilla (Template):** es la que se encarga de mostrar la información a los usuarios a través de un navegador web.
  - **Vista (View):** Es la conexión entre el modelo y la platilla, por consiguiente, es la que determina qué datos se mostrará y por qué plantilla (template).

Funcionamiento MTV de Django

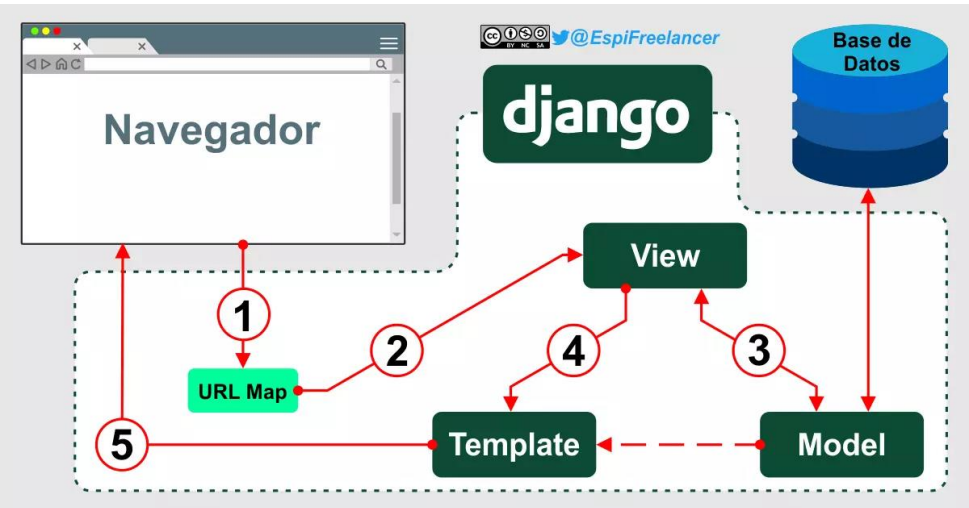


Figura 5. Funcionamiento de MTV de Django [67]

La ilustración anterior (Figura 5), describe de forma gráfica el funcionamiento del patrón de diseño MTV. Básicamente, la lógica del MTV es la siguiente: el usuario a través de un navegador web realiza petición (1), es decir, la View (2), este recibe la petición y pasa al modelo (3) y en este instante el modelo busca información requerida en la base datos. Una vez encontrado la información requerida regresa a la vista (3), y de ahí que pasa al template (4), en este momento se presenta la información que el usuario ha solicitado (5).

Bases de datos

Así como se ha dado a conocer diferentes tipos de lenguajes de programación especializada en Web Scraping, existe también una variedad de plataformas de administradores de bases de datos, sistemas que brindan el servicio de almacenamiento de datos. Cada uno ofreciendo sus distintos funcionamientos, pero siempre cubriendo las necesidades de los usuarios. Estos bases de datos pueden ser tanto relacionales y no relaciones (también conocidas como NoSQL). Las bases de datos SQL o relacionales figuran como:

- ✓ MySQL
- ✓ PostgreSQL
- ✓ SQLite
- ✓ MariaDB
- ✓ Oracle
- ✓ Microsoft SQL Server
- ✓ Neo4j (Este último es un tipo de base de datos orientada a grafo)

Entre los bases de datos NoSQL se puede mencionar los siguientes:

- ✓ Cassandra
- ✓ MongoDB
- ✓ DynamoDB
- ✓ CouchDB

Repositorios de código

Actualmente existen diferentes tipos de repositorios de código de las cuales la opción a elegir depende de cada desarrollador. Como cualquier herramienta, estas plataformas también poseen puntos fuertes y débiles. No obstante, cada una de ellas cuenta con diversos soportes para plataformas de control de versiones. Básicamente, un repositorio de código es un sitio donde el código de una aplicación, de un programa cualquiera está almacenado y desde donde se puede distribuir. Entre los sistemas más conocidos está GitHub, Bitbucket, GitLab, entre otros. De los cuales se hablará a continuación:

GitHub

Es la plataforma más utilizada en el mundo para el control de versiones Git. Una de las razones de la gran fama que tiene Git es el triunfo de GitHub en este ámbito, un sistema web de desarrollo donde los desarrolladores pueden contribuir en uno, varios o determinados proyectos. La plataforma GitHub brinda la mayor función de Git, comprende las distintas herramientas como el control de acceso, colaborativo, trabajabilidad, administración de actividades y la inspección de proyectos. Recientes estudios muestran que pedagogos dentro y fuera del ámbito académico vinculado a la Informática están empezado a utilizar GitHub en sus enseñanzas en el contexto de ingeniería de software. GitHub provee un servicio para albergar repositorios Git de forma remota, esta plataforma fue desarrollado en el año 2008, además provee una interfaz Web donde el usuario inscrito puede crear repositorios ya sean vacíos o a través de la clonación de otros repositorios alojados en GitHub (esta acción o término en GitHub es conocido como *fort*), mandar petición de cambio entre repositorios creados (esta acción o término en GitHub es conocido como *pull request*), y administrar estas peticiones. Los proyectos creados en GitHub son de acceso público de forma predeterminada, por lo que sólo el usuario de pago o en el caso de que cumplan ciertos requisitos, es posible alojar repositorios privados. Así GitHub cuenta con planes de pago que van desde 4,00 dólares por persona al mes (Team), y plan empresarial desde 21,00 dólares por persona al mes (Enterprise). Para el ámbito educativo, vale recalcar que GitHub dispone tipos de cuentas únicas para usuarios como docentes, estudiantes e instituciones educativas. Entre sus

usuarios figuran como: Pinterest, American Airlenes, SAP, Spotify, stripe, Ford, nu bank, Qualcomm [68], [69].

Tabla 6. Aplicación de repositorios públicos de GitHub en la enseñanza [68]

Institución de educación superior	*Tema*	*Aplicación*	*A partir de*	*Link*
U. Chicago (EE.UU.)	Varios cursos	Material de docencia	2012	https://github.com/uchicago-cs
U. California Davis (EE.UU.)	Genómica	Material de docencia	2013	https://github.com/RILAB
HiOA (Noruega)	Varios cursos	Material de docencia	2013	https://github.com/hioa-cs
U Politécnica Madrid (España)	Linked Data	Prácticas	2013	https://github.com/FacultadInformatica-LinkedData
U Complutense de Madrid (España)	Procesadores de Lenguajes	Prácticas	2014	https://github.com/plgucm

La Tabla *6.* Aplicación de repositorios públicos de GitHub en la enseñanza [68]Tabla *6* presenta los cursos que se encuentran disponible en la plataforma GitHub de forma pública. Se calcula que desde el 2014, existía más de 1 200 cursos y 70 000 alumnos usando este tipo de recursos. Por lo que no es de extrañarse el creciente número de usuarios de GitHub actualmente, pues así lo exponen diferentes investigaciones que se han dedicado a explorar esta herramienta, ya sea solamente para estudiar su funcionalidad o para averiguar el número de usuarios de la plataforma.

En este contexto, según la encuesta entre los desarrolladores de Stack Overflow, más del 80% de los desarrolladores usan GitHub. A continuación, se muestra el siguiente gráfico (Figura *6*) con los resultados de la encuesta.

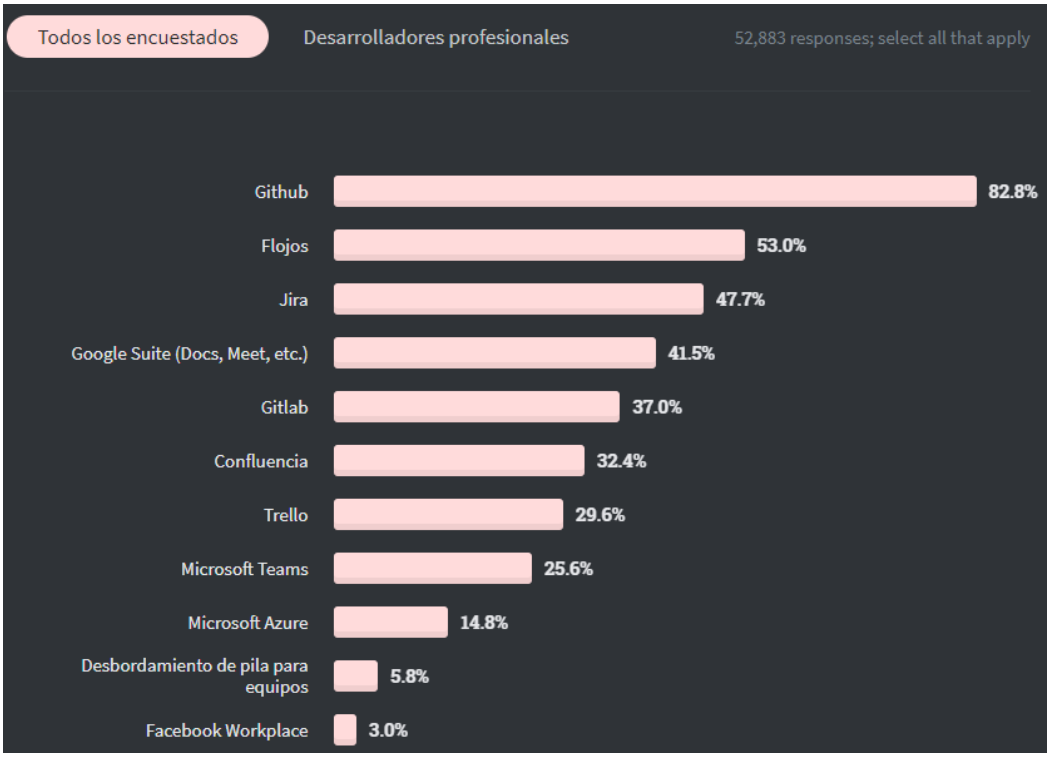


Figura 6. Encuesta Anual de Desarrolladores de Stack Overflow [70]

La Figura *6*, presenta los resultados de la encuesta realizada en este año (2020) a las 65 000 personas desde más de 180 países. Es una encuesta anual para desarrolladores por parte de Stack Overflow que examina todos los aspectos de la experiencia del desarrollador, desde la satisfacción profesional y la búsqueda de empleo hasta la educación y opiniones sobre software de código abierto. La encuesta evidencia que GitHub, básicamente los deja atrás a otras plataformas en este aspecto de versionado de software. Cabe recalcar que la figura anterior es un fragmento de la información que han generado los encuestadores, no obstante, en el web oficial está disponible más información con más ítems de la encuesta.

Bitbucket

Al igual que el anterior, es una plataforma que permite el alojamiento de proyectos que emplean el sistema de control de versiones basados en Mercurial y Git. Tiene plan gratuito para un máximo de 5 usuarios y de pago que van desde 3,00 y 15,00 dólares por usuario al mes (Standard), hasta 6,00 y 30,00 dólares por usuario al mes (Premium). Sus mayores usuarios son Ford, Paypal, Wework, Pandora, entre otros. A diferencia de

GitHub, este repositorio de código permite publicar los códigos de forma privado en su plan gratuito, por lo que para hacer pública los códigos se debe comprar un plan. Según la página oficial, la plataforma cuenta con más de 1 millón de equipos y 10 millones de desarrolladores [71]–[73].

GitLab

Este sistema tiene la versión gratuita conocido como Gitlab CE (*Community Edition*) y una versión empresarial conocido como Gitlab EE (*Enterprise Edition*). Es un sistema para control de versiones de proyectos o de códigos, donde además los desarrolladores pueden contribuir o también denominado grupos de trabajo. Al igual que los anteriores, esta plataforma también cuenta con plan gratuito, y en cuanto a planes de pago rondan entre los 4,00 y 99,00 dólares por persona al mes. Entre sus usuarios están las organizaciones como la NASA, INGO, Goldman Sachs, Bayer, Sony, Freddie, EA, Lockheed Martin, Nomura, Citrix, Siemens, Ask Media Group, U.S. AIR FORCE, Universidad de Washington, Wag, Wish, Worldline y Equinix [74], [75].

SourceForge

Es otra plataforma de repositorios de códigos alternativa a GitHub. Ofrece la opción autenticación multifactor (MFA), un método de seguridad que consiste en utilizar más de una forma para autenticar y con ello comprobar la legalidad de una transacción, lo que genera una alta seguridad [76].

Cloud Source Repositories

Esta plataforma es la encargada de la gestión de Google Cloud Platform. Surgió después del fracaso de Google Code. Este repositorio permite vincular otros repositorios a través de github o Bitbucket según las necesidades. Por ser parte del gigante de Google, ofrece la posibilidad hacer uso de los repositorios de este [77]. Al igual que el sistema Bitbucket, permite almacenar repositorios git privados y de forma ilimitada. Además, permite el despliegue directamente desde el mismo repositorio brindando así la posibilidad de crear y probar el código fuente automáticamente en la misma plataforma.

GitKraken

Es un sistema de repositorios de códigos creado en 2014 por la empresa Axosoft con sede en Arizona, EE.UU. al sistema se le conoce, sobre todo, por integrar una interfaz muy atractiva, por centrarse en la velocidad y por el fácil de manejo de Git. En su versión gratuita es aplicale para una empresa con menos de 20 trabajadores o para organizaciones sin fines de lucro. Soporta las principales plataformas de sistemas como Mac, Windows y Linux [79].

Apache Allura

Este último una implementación de código abierto para el alojamiento de proyectos de código abierto, un sitio web que gestiona repositorios de código fuente, informes de errores, debates, páginas wiki, blogs entre otros. Esta plataforma cuenta una sintaxis de búsqueda avanzada, lo que permite guardar las consultas más habituales [77], [79]

API

La sigla API hace referencia a las siguientes expresiones Application Programming Interface (entendido en Español como Interfaz de Programación de Aplicación). Es un conjunto de funciones, reglas y técnicas que permite comunicar e interactuar entre los diferentes componentes de un software. Esto da una idea de que, una API, básicamente permite la conexión de deferentes programas, siendo este conocido como una interacción "software-to-software". Con los avances en el desarrollo de aplicaciones locales y web existe una cantidad enorme de APIs de diferentes servicios, tal es el caso de Google Map API, Wikipedia API, Chatbot API, Translate API, la lista es enorme, pero a manera de ejemplos se menciona algunos. Siendo estos públicas o privadas, lógicamente, dependiendo del propósito del software y la organización que haya desarrollado [80].

Es así como, en base a lo expuesto, se puede mencionar el API de GitHub, una plataforma de repositorios de códigos.

API GitHub

Como se ha indicado anteriormente, GitHub es el sistema de alojamiento o repositorios de proyectos más utilizado a nivel mundial, cuenta con alrededor de 9 millones de usuarios con cuentas privadas y 17 millones de usuarios públicos. Tales cantidades la ha convertido en el más popular repositorio de códigos basado en la web, incluso siendo objeto de estudio por los diferentes investigadores, asimismo, por los desarrolladores, por ejemplo, hay estudios que prefieren poner en indagación solamente las características del código fuente de los proyectos. Sin duda alguna su prominente característica de la codificación social, las características de los usuarios y la colaboración entre los desarrolladores atrae la atención de los desarrolladores y no desarrolladores. Con todo aquello, se puede indicar que GitHub cuenta con una REST API v3. En su sitio oficial se detallan los procedimientos a seguir para la conexión con este [81].

Se expuso los repositorios de código más utilizados en la web, y el concepto de la API, por lo que cabe indicar que, en este proyecto de investigación, para la extracción de datos, se utilizó una única plataforma como fuente de información, en este caso: GitHub.

Metodología del desarrollo

Para el desarrollo del trabajo, el método optado es el llamado Modelo en Cascada. La misma que sigue un proceso secuencial haciendo que el avance del método tienda ir hacia abajo (como su nombre lo indica como en una cascada de agua) mediante las etapas de análisis de los requerimientos, el diseño, la implementación (desarrollo) y testo (prueba) (Figura 7).

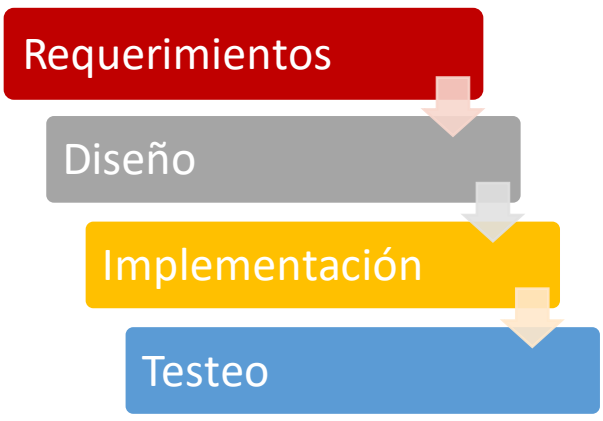


Figura 7: Metodología de desarrollo en cascada [10]

En base la Figura 7 sobre las fases de la metodología en cascada se puede mencionar lo siguiente:

En la *primera etapa* se consideran los requerimientos de los clientes, es decir, esto permitirá definir los objetivos o los alcances del proyecto. De esta etapa sale una documentación con las especificaciones de requerimientos, la misma debe contener toda la especificación sobre lo que debe realizar el sistema, cuidando que no se revele los funcionamientos intrínsecos.

En la etapa de *diseño* se dividen los componentes del sistema en partes, de manera que puedan ser trabajado de forma individual cada componente. De manera que, en esta etapa también surge un documento que recoge todo el diseño del programa, la misma que incluye el detalle de la constitución integral del software y las especificaciones de la función que debe cumplir cada uno de los elementos, y de la forma en que se interactúan entre ellos.

En la etapa de *implementación* o *desarrollo*, se realiza lo que es la programación o también conocido como el código fuente, esta etapa se desarrolla en base los diseños realizados en la fase anterior, al igual que el testeo. Estas actividades se llevan a cabo con el fin de detectar fallas en la programación y corregirlas. Según el lenguaje de programación, la versión y las necesidades del proyecto se usan los módulos. Los desarrolladores tienen la posibilidad de reutilizar las partes de los códigos dentro del mismo proyecto para agilizar el proceso de desarrollo de la programación.

Finalmente, la etapa de *prueba*, donde los componentes que han sido desarrollados se juntan para dar estructura al sistema, una vez hecho esto, se realizan las respectivas pruebas para verificar que el sistema funciona perfectamente. Para ello, se realiza una búsqueda sistemática, en el caso de que exista alguna falla, se depuran estas fallas antes de entregar al dueño del software o cliente. Esta etapa brinda la posibilidad de que el cliente pruebe el sistema, con el que se garantiza que se cumplen sus requerimientos [10].

Fundamentación legal

La investigación realizada tiene su fundamento legal en el Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, en los siguientes

artículos que constan el libro I, título II que hace referencia a los órganos y entidades del sistema nacional de ciencia, tecnología, innovación y saberes ancestrales, capítulo segundo, la misma que prescribe sobre la protección de los derechos intelectuales, a través de los artículos siguientes:

Art. 10.- Autoridad nacional competente en materia de derechos intelectuales. - Es el organismo técnico adscrito a la Secretaría de Educación Superior, Ciencia, Tecnología e Innovación, con personalidad jurídica propia, dotado de autonomía administrativa, operativa y financiera, que ejerce las facultades de regulación, gestión y control de los derechos intelectuales y en consecuencia tiene a su cargo principalmente los servicios de adquisición y ejercicio de los derechos de propiedad intelectual, así como la protección de los conocimientos tradicionales. Además de las funciones inherentes a sus atribuciones, será la principal encargada de ejecutar las políticas públicas que emanen del ente rector en materia de gestión, monitoreo, transferencia y difusión del conocimiento.

La autoridad nacional competente en materia de derechos intelectuales tendrá competencia sobre los derechos de autor y derechos conexos; propiedad industrial; obtenciones vegetales; conocimientos tradicionales; y, gestión de los conocimientos para incentivar el desarrollo tecnológico, científico y cultural nacional. Competencias que deberán ser consideradas al momento de reglamentar su conformación, atribuciones, organización e institucionalidad.

Del mismo modo en el libro II, título II de los derechos de autor y los derechos conexos sección quinta, parágrafo primero del software y bases de datos, apartado primero del software de código cerrado y bases de datos en los siguientes artículos:

Art. 131.- Protección de software. - El software se protege como obra literaria. Dicha protección se otorga independientemente de que hayan sido incorporados en un ordenador y cualquiera sea la forma en que estén expresados, ya sea como código fuente; es decir, en forma legible por el ser humano; o como código objeto; es decir, en forma legible por máquina, ya sea sistemas operativos o sistemas aplicativos, incluyendo diagramas de flujo, planos, manuales de uso, y en general, aquellos elementos que conformen la estructura, secuencia, y organización del programa. Se excluye de esta protección las formas estándar de desarrollo de software.

Art. 132.- Adaptaciones necesarias para la utilización de software. - Sin perjuicio de los derechos morales del autor, el titular de los derechos sobre el software, o el propietario u otro usuario legítimo de un ejemplar del software, podrá realizar las adaptaciones necesarias para la utilización de este, de acuerdo con sus necesidades, siempre que ello no implique su utilización con fines comerciales.

Art. 133.- Titulares de derechos. - Es titular de los derechos sobre un software el productor, esto es, la persona natural o jurídica que toma la iniciativa y responsabilidad de la realización de la obra. Se presumirá titular, salvo prueba en contrario, a la persona cuyo nombre conste en la obra o sus copias de la forma usual.

Dicho titular está además autorizado para ejercer en nombre propio los derechos morales sobre la obra, incluyendo la facultad para decidir sobre su divulgación.

El productor tiene el derecho exclusivo de impedir que terceras personas realicen sin su consentimiento versiones sucesivas del software y software derivado del mismo.

Las disposiciones del presente artículo podrán ser modificadas mediante acuerdo entre los autores y el productor.

Términos Relevantes

SSL

Según [82], SSL es una capa, compuesto de un conjunto de protocolos criptográficos que son usados para proporcionar seguridad en las comunicaciones a través de la red. Desde su aparición en el año 1993, junto con el protocolo HTTPS, ha tenido una evolución a lo largo del tiempo, brindando más seguridad, corrigiendo errores y problemas de implementación y diseño.

Minería de datos

Es una técnica que, por medio del cual se puede descubrir o entender muchas situaciones, ya que esta técnica se basa en patrones, asociaciones, cambios, anomalías,

utilizando enormes cantidades de información ya sea estos bases de datos, almacén de datos u otros sistemas de alojamientos de datos. Este procedimiento implica la utilización de métodos de la Inteligencia Artificial (IA) y la estadística [83]. Esta técnica forma parte del área de investigación que consiste en el desarrollo y aplicación de modelos computacionales que ayudan al descubrimiento de patrones en grandes conjuntos de información [84].

Minería de texto

También llamado como análisis de texto que tiene como objetivo extraer contenido significativo de texto, ya sea en documentos, correos electrónicos o comunicaciones de forma corta como tweets y textos SMS [85], a partir del cual se procede a analizar y descubrir patrones de interés, incluyendo tendencias y anomalías de datos textuales [12].

Big Data

Según [12], big data es el campo multidisciplinario dedicado al análisis de grandes de volúmenes de datos no estructurados. Mientras que [86], considera que es una agrupación de distintas técnicas, facilitando así la manipulación de grandes volúmenes de información procedentes de diferentes orígenes, la misma que debe ser capaz de proporcionar valor. En este sentido, se puede determinar que, big data es un concepto que hace alusión al gran desarrollo que tiene en estos tiempos el acceso a la información y la automatización de este. En otras palabras, los enormes volúmenes de datos digitales manejadas por empresas, instituciones, y entre diferentes entidades, las cuales deben hacer estudios de los datos que generan utilizando o aplicando algún método o algoritmos de big data. Por lo que, en realidad el big data no sólo hace referencia a la tecnología, sino se refiere a la adquisición de valores, lo que significa conocimiento para las compañías o cualquiera otra entidad para mejorar sus actividades.

Bot

Un Bot, básicamente es un programa informático que ha sido dotado para realizar actividades de forma automática, convirtiendo esto en algo recurrente por la vía Internet. Entre los ejemplos más comunes están los exploradores web de los sistemas de búsqueda

de la Internet, por lo tanto son capaces de recorrer las páginas web automáticamente y recopilar datos de las páginas de una forma ágil y eficaz [87], [88]. Además, de estos programas informático, existen otros conocidos como chatbots, que son agentes Inteligentes que tienen la capacidad de comprender un idioma hablado y utiliza la comunicación oral como interfaz de usuario. Estos agentes forman parte de la categoría especifica de software basado en IA. Entonces, vale indicar que es una construcción artificial que está diseñada para conversar con los seres humanos utilizando el lenguaje natural como entrada y salida. Por lo que es ineludible mencionar que, hoy en día, el procesamiento del lenguaje natural es un componente esencial de muchos programas de computadora que buscan interpretar el habla humana [89], [90].

CAPÍTULO II

METODOLOGÍA

Diseño Metodológico

La investigación realizada se basó en lo cualitativo, esto permitió determinar cada uno de los temas en cuestión a través de la recolección de la información, en este caso, fue indagar las distintas técnicas y herramientas de web scraping. Además, el estudio que se realizó fue de tipo exploratorio y descriptivo ya que estos permitieron describir los conceptos de las tecnologías web scraping a través de la búsqueda de la información, así cumplir con el objetivo principal que es desarrollar una aplicación para extraer información desde la API de un repositorio de código utilizando la técnica de web scraping.

En este sentido, la investigación realizada también incluyó el método experimental, ya que se emplearon las técnicas y herramientas necesarias para generar una solución, en este caso, desarrollar una aplicación para extraer información desde las APIS de repositorio de código.

Tipos de investigación

Investigación Descriptiva

La investigación desarrollada es de tipo descriptiva, ya que permitió entender el por qué el uso de las tecnologías Web Scraping actualmente por las organizaciones, por los científicos de datos, investigadores, en el análisis de datos, en el proceso de negocio para la toma de decisiones. Asimismo, permitió indagar sobre las diferentes nociones y significado de cada uno de los tópicos en cuestión.

Investigación exploratoria

La investigación fue también de tipo exploratoria ya que, a través de búsquedas hechas en distintos sitios, plataformas, bibliotecas, repositorios, y revistas virtuales se obtuvo diversos materiales teóricos, siendo esta cada información actualizada. Lo que facilitó conocer los distintos tipos de herramientas existentes para el proceso de web scraping.

Métodos y técnicas

El método experimental

Se utilizó este método a fin de aplicar o emplear las técnicas y herramientas necesarias para generar una solución, así cumplir con el objetivo principal que es desarrollar una aplicación para extraer información desde la API de un repositorio de código utilizando la técnica de web scraping.

El método teórico

El método teórico utilizado en la investigación fue inductivo debido a que se estudió hechos particulares para deducir la problemática general, además se utilizó el método analítico con el fin de comparar las distintas técnicas y herramientas de web scraping existentes en el mercado.

Técnica de investigación

La técnica fundamental utilizada para la recolección de datos es el uso de la aplicación como tal, desarrollada mediante el lenguaje Python, y sobre la cual se llevaron a cabo las respectivas pruebas de rendimiento del software.

Población y muestra de estudio

Teniendo en cuenta la naturaleza de la investigación realizada, los datos estadísticos correspondientes a la población y muestra sobre un universo de estudio determinado no se emplean en este contexto.

Descripción y validación del instrumento

Conociendo el sentido del estudio que es desarrollar un software que sea capaz de extraer información desde repositorio de código, por lo cual, el principal instrumento o herramienta es la propia aplicación. La cual fue desarrollada con el fin de soportar grandes cantidades de extracción de la información desde la plataforma de alojamiento de código y su correspondiente análisis. Por lo que es referida como Web Scraping (o en este caso se llama ScrapGit, nombre que se le ha asignado al Software), básicamente, el software debe brindar la información requerida por el usuario, por ejemplo, títulos, links, otros ítems como números de commit, número de colaboradores, fecha de publicación de repositorios, así como también lenguajes de programación utilizados en los proyectos, pues esto permite una mejor compresión sobre la evolución actual del software y de las herramientas actuales de desarrollo y proyectos como tal.

Especificación de requisitos

ScrapGit es una aplicación web y cuenta con características básicas de un buscador. La aplicación fue desarrollada bajo el framework Django cuyo lenguaje de programación es Python y la base de datos con la cual funciona es MongoDB, herramientas que en su conjunto forman el denominado Software para Web Scraping (Figura *8*).

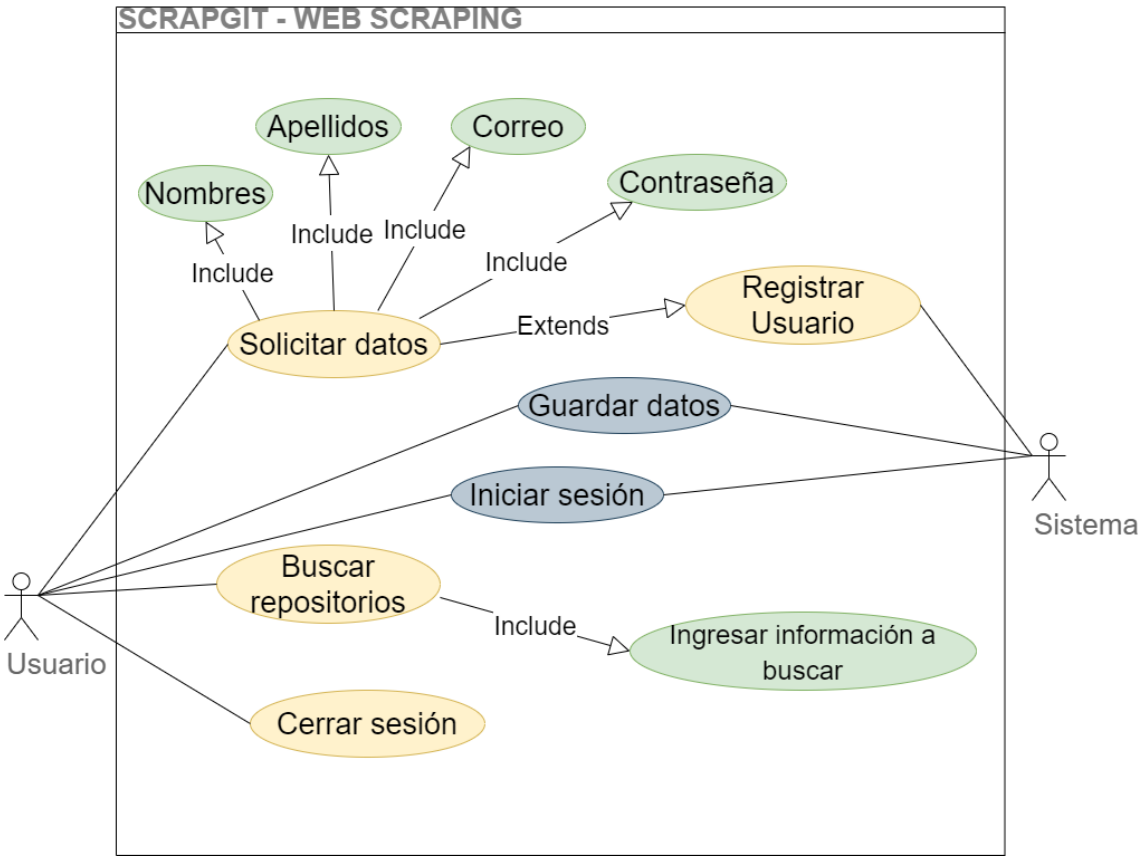


Figura 8. Diagrama de Casos de Uso de ScrapGit
Fuente: Elaboración propia

Dentro de los requisitos funcionales en las especificaciones de requerimientos de software se puede mencionar lo siguiente:

- ✓ Ver Páginas Web Scrapeadas
- ✓ Ver detalles (items) de las webs scrapeadas
- ✓ Acceder al repositorio por medio del enlace
- ✓ Buscar repositorios

Mientras tanto, dentro de los requisitos no funcionales se presenta lo siguiente:

- ✓ Rendimiento
- ✓ Seguridad
- ✓ Fiabilidad
- ✓ Portabilidad
- ✓ Mantenimiento

Tabla 7. Modelo Requerimiento funcional

Identificación de requerimiento	RF01
Nombre del requerimiento	Ver páginas web scrapeadas
Descripción del requerimiento	La aplicación debe permitir ver repositorios scrapeados
Prioridad del requerimiento:	Alta

Fuente: Elaboración de propia

La Tabla *7* es una descripción de modelo de unos de los requerimientos funcionales del software con las características de una Historia de Usuario según el estándar IEEE 830 de Especificación de Requerimientos de Software (ERS).

Tabla 8. Modelo Requerimiento No Funcional

Identificación de requerimiento	RNF01
Nombre del requerimiento	Rendimiento
Descripción del requerimiento	Garantizar que las búsquedas u otro proceso no afecte al desempeño de la DB, ni al tráfico de red, Servidores Web.
Prioridad del requerimiento:	Alta

Fuente: elaboración propia

La Tabla *8* es una descripción de modelo de unos de los requerimientos no funcionales del software con las características de una Historia de Usuario según el estándar IEEE 830 de Especificación de Requerimientos de Software (ERS).

Arquitectura de la aplicación

La aplicación en su forma general consta de dos partes: un *backend* que permite scrapear, es decir, obtener información con la API (hace de interfaz entre la aplicación y la plataforma GitHub) desde el repositorio de código. Además, cumple la tarea de guardar en la base de datos la información extraída. Un *frontend* para realizar búsquedas y visualizar a través de una tabla los resultados (repositorios scrapeadas) a través del

navegador web. La imagen siguiente (Tabla 9Figura 9) muestra un diagrama de la arquitectura de la aplicación.

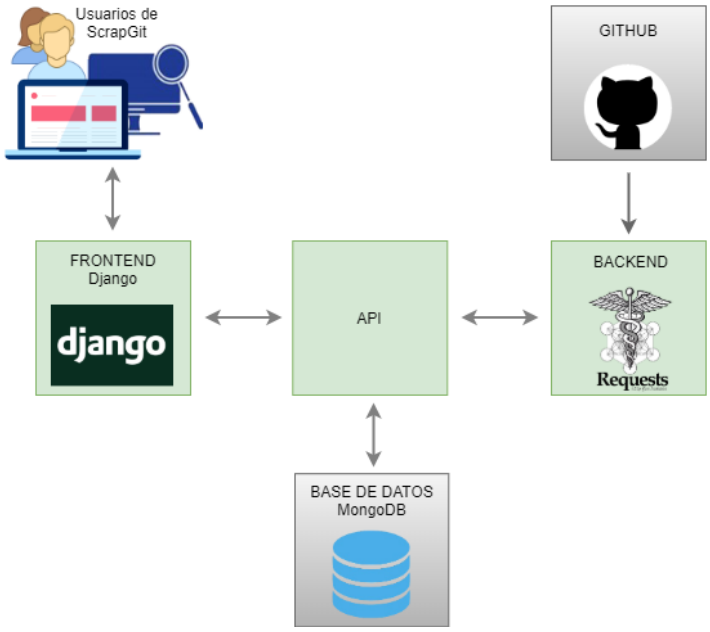


Figura 9. Arquitectura del Software ScrapGit
Fuente: Elaboración propia

Diagrama de flujo

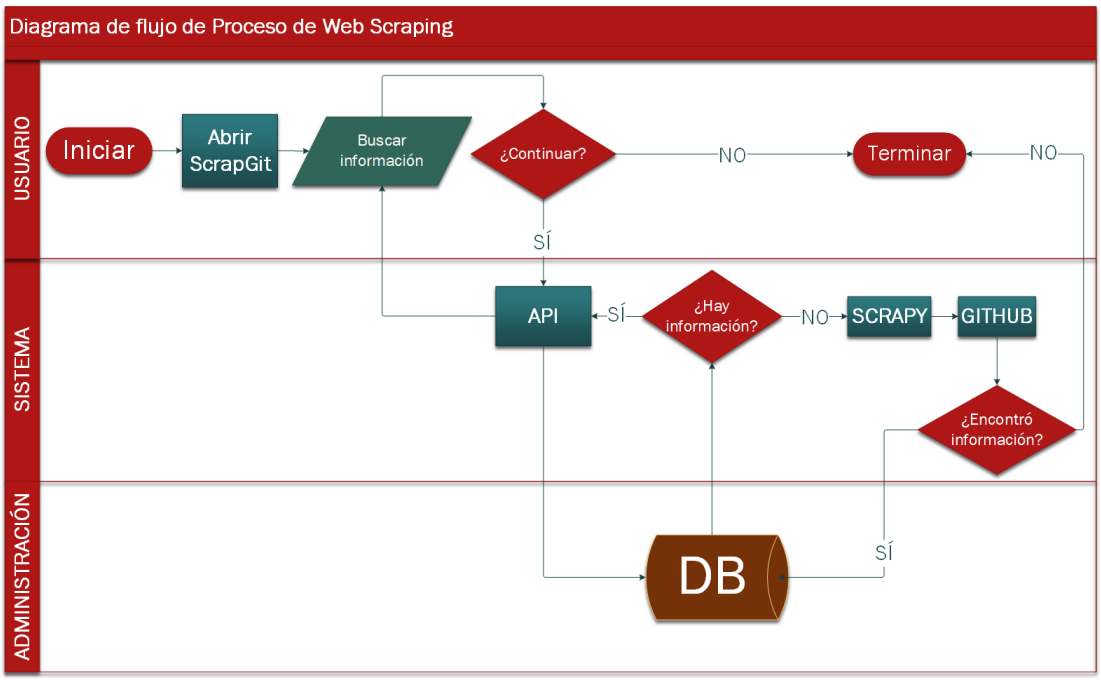


Figura 10. Diagrama de Flujo de Proceso de ScrapGit (Web Scraping)
Fuente: Elaboración propia

Técnicas de procesamiento y análisis de datos

Análisis de la información recolectada

Durante todo el proceso de investigación, indagación, recolección de la información en distintas plataformas virtuales; se obtuvieron datos muy relevantes que permitieron llegar a la solución deseada, entre las cuales fueron: bibliotecas o librerías, framework para distintos lenguajes de programación (que gracias a esta investigación se pudo conocer los distintos lenguajes aplicables en el desarrollo de la aplicación para la web scraping), APIs, entre otros. Asimismo, se conoció también los programas que ya existen para web scraping, no obstante, como se ha señalado anteriormente, estas herramientas dependiendo el uso que se quiera dar, el costo varía, pero no dejar de ser un obstáculo su coste para alguien que no tenga recursos para adquirir con funcionalidades completas, el costo de estos softwares se puede observar en esta tabla. En este sentido en el siguiente apartado se muestra las herramientas elegidas para llevar a cabo el estudio, por ende, el desarrollo del software Web Scraping.

Elección de las herramientas a utilizar

En base la tabla: Librerías y frameworks de web scraping de código abierto y la tabla: Matriz comparativa de lenguajes, después de haber estudiado y analizado cada uno de los lenguajes de programación y cada una de las bibliotecas disponibles para cada lenguaje, y según las aplicaciones de cada herramienta y viendo las facilidades que brinda cada uno en el contexto del desarrollo de aplicaciones para Web Scraping, se optó por utilizar como lenguaje principal de desarrollo Python, se eligió este lenguaje porque es el más utilizado en varios ámbitos como el desarrollo de programas basadas en la web, gestión de sistemas, ciencias de datos, ciencia computacional, inteligencia artificial, Internet de las cosas, entre otros. Además, cuenta con una alta gama de librerías estandarizadas y una gran cantidad de módulos. En este sentido existen bibliotecas especializadas para Python que permiten realizar web scraping, este es el caso de Beautiful Soup, Scrapy, entre otros. De la misma manera, la biblioteca utilizada fue Requests como extractor principal de información a través de repositorio de código. Hay que indicar también que el principal repositorio de código utilizado fue GitHub, y para el

almacenamiento de la información, el semiestructurado MongoDB y como parte de la interfaz de usuario (Frontend) se utilizó el framework Django (Tabla 9).

Tabla 9: Herramientas elegidas para usar en el desarrollo del Software

Lenguaje de programación	Biblioteca	Base de datos	Repositorio de código	Frontend
Python	Requests, API GitHub	MongoDB	GitHub	Django

Fuente: Elaboración propia.

La Tabla 9, muestra cada una de las herramientas elegidas entre algunos de los que se pudieron recabar durante la búsqueda de la información para ser utilizadas en el desarrollo del software.

Con todo lo anterior, para conocer su correcto funcionamiento en forma general, y su rendimiento en cuanto a las peticiones y carga de información, fue necesario realizar distintos tipos de pruebas de software con el propósito de analizar los puntos críticos de la aplicación. Con ello tomar medidas necesarias para dar solución al fallo, en caso de que existiese.

Desde el punto de vista de Mera [91], es factible segmentar los tipos de pruebas de software en tres clases generales; Pruebas Funcionales, las mismas se basan en la interoperabilidad de las operaciones que realiza un determinado sistema, Pruebas No Funcionales, las cuales corresponden a la medición de las características de una aplicación que puedan ser cuantificadas a través de una especificación de escala, y Pruebas Estructurales, estas se focalizan en los procedimientos del software, desde su propio código fuente, verificando los posibles flujos de ejecución en los procedimientos mediante diversas entradas que produzcan salidas controladas.

Conociendo la naturaleza, las características de la aplicación (ScrapGit), las clases de pruebas a considerar para la evaluación corresponde a las pruebas no funcionales, y en relación con las características fundamentales en aplicaciones de Web Scraping, las siguientes pruebas se destacan como principales métodos para analizar el funcionamiento y los aspectos de la aplicación desarrollada con Django en Python.

Tabla 10. Pruebas de Software No Funcionales a realizar [91]

PRUEBA	OBJETIVOS
De Carga	Es tipo de prueba cuyo objetivo es medir el rendimiento del software en diferentes contextos. Las pruebas de carga tienen por objetivo simular demanda sobre una aplicación de software y medir el resultado. Estas pruebas realizan bajo el concepto demanda esperada y también en condiciones que puedan producir una sobrecarga, con el fin de observar cómo responde la aplicación e n diferentes situaciones.
De Estrés	Es un tipo de prueba que por lo general busca determinar la solidez de la aplicación al tratar de llegar al límite de la capacidad operativa tanto la aplicación misma, como en la base de datos. Este tipo de prueba de software se utiliza para determinar la estabilidad de un sistema o aplicación, con especial atención en la disponibilidad y manejo de errores cuando se enfrenta a la sobrecarga.
De Tiempo de Respuesta	El objetivo de este tipo de prueba consiste en medir el tiempo de respuesta de la aplicación sobre los distintos tipos de operaciones y consultas a los que van a estar sometido, dependiendo asimismo de la cantidad de registros a los cuales tiene que acceder.

La Tabla 10, presenta una explicación sobre las definiciones de los tipos de pruebas no funcionales que se llevaron a cabo para evaluar el funcionamiento de la aplicación.

En este marco, para realizar las respectivas pruebas se tomaron en cuenta las siguientes herramientas de testing, todas con el fin de llevar a cabo las pruebas descritas anteriormente y haciendo hincapié en el rendimiento general del software:

- SonarQube
- SoapUI
- FunkLoad
- JMeter
- WebLoad
- Selenium

Después de que se hayan estudiado las herramientas, y tomando en cuenta los tipos de pruebas necesarios para detectar los puntos críticos en la aplicación se contempló usar SonarQube como principal herramienta para llevar a cabo la ejecución de las pruebas, puesto que cuenta con las funcionalidades necesarias para obtener los datos que permitan realizar un análisis a nivel no funcional con el cual tener una visión más integra del rendimiento del software desarrollada. En este sentido SonarQube fue complementada con la herramienta Selenium para simular la concurrencia de usuarios durante la prueba de carga y la prueba de interés. Una vez obtenidos los resultados requeridos mediante la aplicación de las pruebas, estos fueron analizados bajo la perspectiva de caja negra y dentro del marco referencial correspondiente a las métricas relacionados al Comportamiento Temporal y Cumplimiento de la Eficiencia del Software, según las características de Eficiencia de acuerdo con el estándar ISO/IEC 25010, norma que provee parámetros para determinar la calidad del Software.

Normas éticas

Durante todo el proceso de investigación y como futuro profesional doy fe de no incurrir en el plagio, de la misma manera hacer uso correcto de los datos obtenidos durante la investigación y hacer debidas citas mediante herramientas de referencias bibliográficas. Con responsabilidad y ética profesional, conservar la autenticidad de la información aquí sujeta y mantener la privacidad de los datos que puede ser utilizados en el transcurso del estudio y prueba del programa. Pues, además, hacer web scraping implica responsabilidad, es decir, dar crédito de dónde se obtuvo la data o información. Incluye a esto la tarea de verificar la legalidad de los datos antes de su publicación, si así fuere el caso.

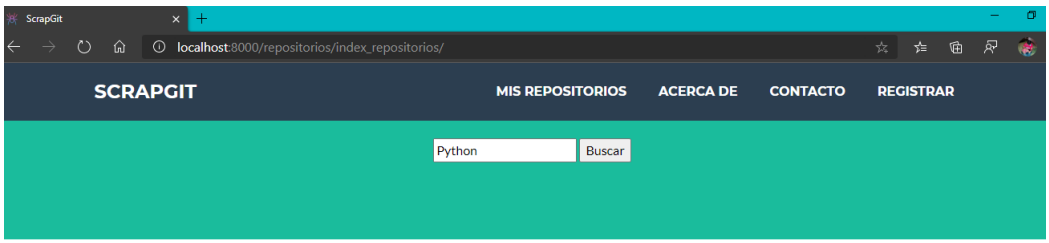
CAPÍTULO III

RESULTADOS

APLICACIÓN WEB PARA WEB SCRAPING DESDE REPOSITORIO DE CÓDIGO GITHUB

Cumpliendo con los objetivos planteados en esta investigación, se llevó acabo el desarrollo de una aplicación web basada en las técnicas de web scraping mediante el uso de API de GitHub, Python como lenguaje de programación, y Django como el marco de desarrollo. Consiste en una página web y es de tipo buscador, por la cual través de una tabla HTML muestra los resultados de la búsqueda de repositorios ya almacenados en la base de datos del software.

En este contexto la aplicación permite la visualización de títulos, descripción, star (favoritos), fork (copia exacta), fecha creación y actualización, commits (URL) entre otros ítems de repositorios de GitHub (Figura 11).



Repositorios Scrapeadas

Título	Descripción	Star	Fork	Fecha Creación	Fecha Actualización	URL Commits	URL Colaboradores	URL lenguajes	URL Repositorio
eclemma	:warning: OLD EclEmma repository, new is located at https://github.com/eclipse/eclemma	84	33	2012-07-08T21:47:35Z	2020-06-01T13:46:42Z	Ver Commits	Ver Colaboradores	Ver Lenguajes	Ver Repositorio
eclemma.com	:bookmark: website https://www.eclemma.com/	0	0	2016-06-04T01:14:59Z	2020-07-15T10:19:51Z	Ver Commits	Ver Colaboradores	Ver Lenguajes	Ver Repositorio
eclemma.org	:bookmark: website https://www.eclemma.org/	0	0	2016-06-04T01:25:52Z	2020-07-15T10:20:01Z	Ver Commits	Ver Colaboradores	Ver Lenguajes	Ver Repositorio
eclipse eclemma	:waning_crescent_moon: Eclipse EclEmma project repository	14	0	2016-10-31T14:09:42Z	2020-08-18T18:34:22Z	Ver Commits	Ver Colaboradores	Ver Lenguajes	Ver Repositorio

Figura 11. Vista de la aplicación desarrollada.
Fuente: Aplicación ScrapGit

Figura 12. Vista de Administrador de Django
Fuente: Framework Django de Python

La Figura 12, muestra la vista del administrador del framework Django de Python. En la imagen se puede observar que existen dos apartados las cuales son **Autenticación y Autorización** la misma que tiene como subapartado *Grupos* y *Usuarios* (por defectos). En los usuarios se encuentran los administradores de la aplicación. En el apartado **Repositorios** (creación propia) se encuentran subapartados como *Repositorios* y *Usuarios*. En el caso de subapartado *Repositorios* se encuentran todos los repositorios scrapeados o guardados en la base de datos.

A nivel estructural, la aplicación fue desarrollada bajo el paradigma Modelo - Template - Vista (MTV) de Django tanto en el backend como en el frontend, de modo que es posible identificar diferentes directorios sobre los cuales se encuentran los respectivos modelos vistas y templates de la aplicación. Siendo los principales directorios "WebScraping/" y "repositorios/" y como subdirectorios "migrations/" y "templates/" (Figura 13).

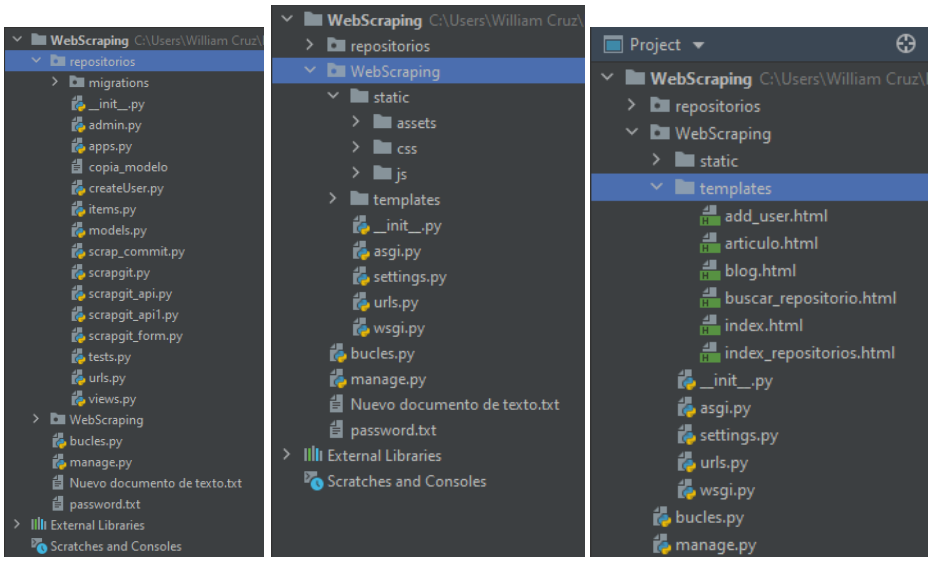


Figura 13. Estructura de la aplicación ScrapGit
Fuente: Aplicación ScrapGit

En base la Figura 13, se puede hacer referencia a las características del proceso de funcionamiento del software según sus directorios de la siguiente manera:

- El directorio “repositorios/” abarca los archivos modelos, admin, y vistas del tipo .py de Python. Además, contiene el archivo scrapgit que realiza peticiones http a través del API de GitHub para realizar el proceso de Web Scraping, es decir, la que extrae repositorios de la plataforma y lo guarda en la base de datos. También maneja el directorio “repositorios/migrations” la cual maneja todos los cambios que se realizan en el modelo, por consiguiente, la base de datos.
- Hay que indicar que, en Python cuando se crea un proyecto se crea un directorio igual al directorio del proyecto (en este caso es WebScraping), razón por la cual existe dos directorios con el mismo nombre. Con esta explicación, hay que mencionar que en los directorios “WebScraping/static” se encuentran los archivos javascript, css, las imágenes e iconos usado en la aplicación.
- Los directorios “WebScrapgin/templates” contiene todas las vistas HTML, es decir, las interfaces de usuario.
- Asimismo, el directorio “WebScraping/” contiene los archivos urls y settings. En urls se definen las rutas a las cuales deberá acceder el frontend para las peticiones que necesite realizar. En settings se configura los datos de la base de datos para su respectivo funcionamiento.
- Finalmente, el directorio del proyecto “WebScraping/” contiene el archivo manage.py, la cual permite realizar actividades como poner en marcha el servidor de Django, realizar cambios en la base de datos (migratios), cambios en el modelo, entre otros.

En cuanto a la base de datos se refiere, como se mencionó anteriormente, el desarrollo de la aplicación ScrapGit funciona junto con la base de datos NoSQL MongoDB, sobre esto se guardan las colecciones que hacen referencia a los principales modelos de la aplicación definidos tanto en el backend como en el frontend, así como también las diferentes configuraciones predeterminadas de Django.

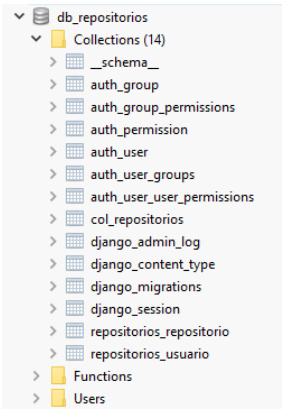


Figura 14. Principales colecciones de la base de datos
Fuente: Base datos MongoDB

Key	Value	Type
(1) ObjectId("5f42f96d97c9f1acdd9a5c60")	{ 11 fields }	Object
_id	ObjectId("5f42f96d97c9f1acdd9a5c60")	ObjectId
titulo	eclemma	String
descripcion	:warning: OLD EclEmma repository, new is located at https://gith...	String
fecha_actualizacion	2020-06-01T13:46:42Z	String
fecha_creacion	2012-07-08T21:47:35Z	String
fork	33	Int32
star	84	Int32
url_colaboradores	https://api.github.com/repos/jacoco/eclemma/contributors	String
url_commits	https://api.github.com/repos/jacoco/eclemma/commits	String
url_lenguajes	https://api.github.com/repos/jacoco/eclemma/languages	String
url_repositorio	https://github.com/jacoco/eclemma	String

Figura 15. Ejemplo de un documento de la colección de "repositorios.repositorio" en la base de datos.
Fuente: Base datos MongoDB

Para el despliegue del software se usó un servidor local Django corriendo con el sistema operativo Windows la misma que cuenta con las siguientes características:

- ✓ 8 GB en memoria RAM
- ✓ 4 CPU's
- ✓ Disco duro de 250 GB

Los datos que han sido almacenadas en la base de datos fueron generados de forma estructurada, es decir, como se trata de repositorios de GitHub, primero se extrajo datos de un determinado usuario de GitHub, a través del API en formato JSON (ver Figura 16 y Figura 17), una vez obtenido la información se determinó qué datos almacenar en la base de datos.

```
{
    "id": 78764436,
    "node_id": "MDEwOlJlcG9zaXRvcnk3ODc2NDQzNg==",
    "name": "chrome-app-samples",
    "full_name": "Josheriff/chrome-app-samples",
    "private": false,
    "owner": {
        "login": "Josheriff",
        "id": 18023518,
        "node_id": "MDQ6VXNlcjE4MDIzNTE4",
        "avatar_url": "https://avatars0.githubusercontent.com/u/18023518?v=4",
        "gravatar_id": "",
        "url": "https://api.github.com/users/Josheriff",
        "html_url": "https://github.com/Josheriff",
        "followers_url": "https://api.github.com/users/Josheriff/followers",
        "following_url": "https://api.github.com/users/Josheriff/following{/other_user}",
        "gists_url": "https://api.github.com/users/Josheriff/gists{/gist_id}",
        "starred_url": "https://api.github.com/users/Josheriff/starred{/owner}{/repo}",
        "subscriptions_url": "https://api.github.com/users/Josheriff/subscriptions",
        "organizations_url": "https://api.github.com/users/Josheriff/orgs",
        "repos_url": "https://api.github.com/users/Josheriff/repos",
        "events_url": "https://api.github.com/users/Josheriff/events{/privacy}",
        "received_events_url": "https://api.github.com/users/Josheriff/received_events",
```


Figura 16. Información del repositorio de GitHub del usuario Josheriff
Fuente: Aplicación ScrapGit

```
col.update_one(
    {
      "titulo": titulo
    # "descripcion": descripcion
    }, {
        "$set": {
            "descripcion": descripcion,
            "star": star,
            "fork": fork,
            "fecha_creacion": fecha_creacion,
            "fecha_actualizacion": fecha_actualizacion,
            "url_commits": url_commits,
            "url_colaboradores": url_colaboradores,
            "url_lenguajes": url_lenguajes,
            "url_repositorio": url_repositorio,
            "titulo": titulo
        }
    }, upsert=True)  # upsert --> Operación de INSERCIÓN en caso de no existir un documento
                        # que cumpla con mi condición. Y Operación de ACTUALIZACIÓN en caso de que
                        # exista un documento que cumpla con mi condición.
```

Figura 17. Proceso de almacenamiento de la información del repositorio en la base de datos
Fuente: Aplicación ScrapGit

ANÁLISIS DE COMMITS

Una vez almacenado los repositorios en la base de datos, se procede a la consulta de esta en el que se puede apreciar los ítems de los repositorios (Figura 11). En base la consulta se puede hacer las siguientes acciones, por ejemplo, ver commits (a través de hipervínculo), ver número de colaboradores, lenguajes de programación utilizado en el proyecto, entre otros. Es necesario enfatizar que en este apartado se analiza los commits realizados en cada proyecto, en este sentido, revisando cada uno de los repositorios

scrapeados se observa que los proyectos alojados en la plataforma de Github dependiendo del alcance que tienen cada uno el número de commits es elevado o poca (al igual que el número de colaboradores). En este sentido, sólo revisando los commits se puede evidenciar o darse cuenta de que un determinado proyecto tiene mucha relevancia y lo mejor es que el proyecto está recibiendo mantenimiento, soporte y/o actualización. Además de esto también se puede tomar en cuenta los números colaboradores.

Dicho lo anterior, se presenta la siguiente Tabla 11 en el que se muestra un resumen de ítems de los proyectos scrapeados desde GitHub. En este sentido la tabla muestra título, star (favorito), fecha de creación, fecha de actualización, colaboradores y lenguajes de programación (utilizados en el proyecto). En resumen, el proyecto 1 tiene un 84 marcado como favorito, fue creado en julio de 2012 y última fecha de actualización junio 2020, tiene 7 y utiliza 4 lenguajes de programación. El proyecto 2, tiene marcado 24 como favorito, creado el octubre de 2016 y última actualización agosto de 2020, tiene 5 colaboradores y utiliza 4 lenguajes de programación. Continuando con el resumen, el proyecto 3, tiene 2306 personas que han marcado como favorito, fue creado julio de 2012 y actualizado en agosto de 2020, cuenta con 30 colaboradores y tiene 9 lenguajes de programación. Finalmente, el proyecto 4, tiene 17 favoritos, creado en enero de 2017 con fecha de última actualización febrero de 2020, cuenta con 30 colaboradores y utiliza un único lenguaje de programación que es el JavaScript.

Tabla 11. Resumen de análisis de Commits

#	Título	Star	Fecha creación	Fecha actualización	Colaboradores	Lenguaje
1	eclemma	84	2012-07-08	2020-06-01	7	Java, HTML, Python, Shell
2	eclipse eclemma	14	2016-10-31	2020-08-18	5	Java, HTML, Shell, CSS
3	JaCoCo - Java Code Coverage Library	2306	2012-07-08	2020-08-23	30	Java, HTML, JavaScript, Kotlin, XSLT, Groovy, CSS, Scala, Shell
4	clean code javascript	17	2017-01-12	2020-02-15	30	JavaScript

Fuente: Elaboración propia

En base este resumen se puede indicar lo siguiente: gracias al software Web Scraping se quita la tarea de buscar y rebuscar proyectos con relevancia en GitHub, pues permite obtener de forma autónoma la información sobre los repositorios y con ello sin mucha pérdida de tiempo se puede saber qué proyectos puede servir para replicar, para mejorar o para colaborar en el desarrollo de este. Y lo más importante, se puede saber en qué fecha han sido creados y si estos están siendo actualizados o no, y también se puede saber el número de colaboradores y número de favoritos, las cuales indican lo malo o bueno o lo relevante que es el proyecto alojado en el repositorio de GitHub.

CAPÍTULO IV

DISCUSIÓN

TÉCNICAS Y HERRAMIENTAS DE WEB SCRAPING

El uso de las técnicas de Web Scraping, para extraer información de forma automática a través de las páginas web o APIS, es una ventaja que brindan las diferentes herramientas de este ámbito, ya que no requieren una inversión de grandes sumas de dinero para obtener y hacer uso de ellas. En este sentido, el uso de Web Scraping es una de las ventajas para la obtención de datos de forma automática de las webs. Además, hacer uso de esta técnica es un método que actualmente va tomando fuerza ya sea a nivel educativo o empresarial. Como muestra, a través de este de trabajo y durante la recolección de las distintas informaciones se ha podido observar que el Web Scraping es muy utilizado en la investigación ya sea para proyectos de pregrado o postgrado.

Tal es el caso de "Implementación de algoritmo en el Lenguaje R para extraer los datos de los Perfiles en Google Scholar utilizando la técnica web Scraping de Minería de datos". En este estudio utilizan como lenguaje de desarrollo R aplicando la técnica de Minería de datos. En base lo anterior se pueden indicar que Web Scraping es un método que realmente aporta mucho valor en el estudio de los datos a nivel empresarial permitiéndoles tomar decisiones para la mejora de los servicios o productos según la actividad a la que se dedique la organización. Además, como se puede darse cuenta Web Scraping tiene mucha relación con otras técnicas como es la inteligencia de negocio y big data.

Por otra parte, el "Desarrollo de un buscador de recetas basado en web scraping", es otro estudio en el que se hace uso de Web Scraping para un proyecto de postgrado de Mcs. El estudio muestra una serie de herramientas que se pueden usar para el desarrollo de la aplicación de este ámbito.

Como muestra el "Desarrollo de un sistema de seguimiento para Instagram", es un ejemplo que demuestra que Web Scraping que puede ser usado en cualquier ámbito o plataformas para obtener datos de ellas. en este estudio, muestra las herramientas como MongoDB, como herramienta para el almacenamiento de la información una vez sean extraídas de la Web o plataformas virtuales.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

Una vez culminada el proyecto de investigación se concluye lo siguiente:

Dentro de las distintas técnicas y herramientas para extraer información desde las APIS de repositorios de código se destaca el web scraping, la misma que puede ser aplicado de varias formas, dicho de otro modo, se puede utilizar los programas ya creados por otras personas; no obstante, estos programas carecen de muchas funcionalidades cuando son gratuitas y cuando son de pago, los precios son muy excesivos. En este sentido, la otra opción es desarrollar una aplicación, el cual es el objetivo principal de este estudio. Cuando se trata de desarrollar una aplicación propia para realizar web scraping, entran en juego dos aspectos muy importantes para tener en cuenta, es decir, el software puede ser desarrollado ya sea utilizando librerías o framework o utilizando el API distribuido por el sitio objetivo. A nivel de librería o framework, el más utilizado en el ámbito de web scraping es el Scrapy, una herramienta que para el proceso de extracción de la información utiliza XPATH o Expresión regular (regex por su abreviatura en inglés), en el cual se especifican los ítems que serán extraídos del sitio web, además está desarrollado para el lenguaje Python. A nivel del API, básicamente se utilizó la documentación proporcionada por el sistema o sitio web, el cual muestra los distintos métodos para hacer web scraping desde la API. Única librería requerida es el requests. Es necesario hacer hincapié que todo este proceso fue aplicada en una única plataforma de repositorios de código (GitHub).

Una vez recolectada suficiente información sobre las distintas técnicas se procedió a realizar un análisis comparativo de estas técnicas con el fin de elegir la más adecuada para desarrollar el software web scraping. Por ejemplo, la Tabla 2, Tabla 3, Tabla 4 y Tabla 5 muestran cada una de las herramientas existente para web scraping así como librerías, APIs, lenguajes de programación, programas web scraping, entre otros. Tabla 9 presenta las herramientas elegidas para desarrollar la aplicación.

Después de que se ha realizado el análisis comparativo de las herramientas, y con ello haber seleccionado las herramientas actuales más apropiadas se procedió a

desarrollar el software web scraping denominado ScrapGit. Cada de una de las herramientas utilizadas en este estudio actualmente son muy aplicadas en el ámbito de desarrollo de software bajo la técnica de web scraping, por ejemplo, el lenguaje Python, aunque no es muy actual, pero debido a que este es mejorado y optimizado en cada actualización y además teniendo en cuenta que este lenguaje posee la más alta gama de módulos, preferido por los desarrolladores para la extracción de datos y para los científicos de datos para el análisis estadístico, se eligió usar para desarrollar el programa. Junto a este lenguaje, para el desarrollo de la interfaz de la aplicación se utilizó el framework Django, el cual es un marco de desarrollo que sirve para crear aplicaciones web. Este framework al igual que Python es mejorado en cada lanzamiento.

Con todo ello, el software ScrapGit puede extraer información desde el api de repositorio de código GitHub, luego, almacenarlo y mostrarlo mediante plantillas según la búsqueda realizada por el usuario. En este sentido los objetivos propuestos al inicio del estudio fueron cumplidas a toda cabalidad.

RECOMENDACIONES

Una vez concluida el estudio se recomienda lo siguiente:

Es necesario tener muy clara la problemática del estudio y con ello definir bien los objetivos generales y específicos, pues estos son determinantes sobre la dirección que debe seguir el proyecto, además de permitirle saber y entender los recursos que se utilizará para cumplir con los objetivos planteados, valga la redundancia.

Es fundamental realizar búsqueda exhaustiva de la información en distintas revistas, bibliotecas o plataformas virtuales con la finalidad de recabar la mayor cantidad de datos posibles, esto permite sustentar bien cada uno de los temas a tratar, con ello también enriqueciendo el conocimiento. En esta búsqueda de la información es posible que se hayan datos no relacionados con el tema principal del estudio, por el cual es recomendable hacer cuadros o tablas comparativas, incluyendo las pruebas, para así elegir las herramientas más acordes posibles al proyecto.

En cuanto a la parte técnica, es imprescindible conocer buenas prácticas con respecto al desarrollo de código, y tener un paradigma de programación definido (muchas veces

esto depende del framework que esté utilizando), por ejemplo, en este caso, el paradigma empleado es el MTV; debido a que el framework utilizado es el Django (para el desarrollo web en el que se involucran diferentes herramientas que dan estilos a la página web en sí) junto con el lenguaje Python. En este sentido, teniendo en cuenta la variedad de herramientas actuales de desarrollo disponible para cada lenguaje de programación, es recomendable estudiar bien estas antes de poner en marcha el proyecto, además depende del enfoque que se quiere dar a la aplicación a desarrollar, es decir, si el software a desarrollar está enfocado en la web, será necesario utilizar herramientas basada en la web. Siguiendo esta idea, es recomendable tener muy en cuenta a las herramientas actuales de desarrollo; ya que estos brindan mayor optimización, mejoras, compatibilidad, configuración, seguridad y facilidad de uso, etc.

REFERENCIAS

[1] J. Sayago, "Máster Universitario de Investigación en Ingeniería de Software y Sistemas Informáticos," 2017.

[2] S. Kemp, "Digital 2019: Global Internet Use Accelerates ," *We Are Social Singapore*, 31-Jan-2019. [Online]. Available: https://wearesocial.com/sg/blog/2019/01/digital-2019-global-internet-use-accelerates. [Accessed: 27-Apr-2020].

[3] ITU.INT, "Las estimaciones mundiales y regionales de TIC de 2018," *Comunicado de prensa*, 07-Dec-2018. [Online]. Available: https://www.itu.int/es/mediacentre/Pages/2018-PR40.aspx. [Accessed: 27-Apr-2020].

[4] A. Amalia, R. Maulidya Afifa, and H. Herriyance, "Resource Description Framework Generation for Tropical Disease Using Web Scraping," *2018 IEEE Int. Conf. Commun. Networks Satell.*, pp. 44–48, 2018.

[5] M. D. Rey Suárez, "Componente de extracción y almacenamiento de datos de una red social para una herramienta Web," Unviversidad Católica de Colombia, 2017.

[6] R. S. Chaulagain, S. Pandey, S. R. Basnet, and S. Shakya, "Cloud Based Web Scraping for Big Data Applications," *Proc. - 2nd IEEE Int. Conf. Smart Cloud, SmartCloud 2017*, pp. 138–143, 2017.

[7] D. Blazquez, J. Domenech, J. A. Gil, and A. Pont, "Monitoring e-commerce adoption from online data," *Knowl. Inf. Syst.*, 2018.

[8] D. Murillo and D. Saavedra, "Implementación de algoritmo en el Lenguaje R para extraer los datos de los Perfiles en Google Scholar utilizando la técnica web Scraping de Minería de datos," in *Universidad Tecnológica de Panamá*, 2017, vol. 14, no. 1, pp. 1–10.

[9] J. S. Canós, "Desarrollo de un sistema de seguimiento para Instagram," Universidad Politécnica de Valencia, 2017.

[10] M. C. López Meseguer, "Desarrollo de un buscador de recetas basado en web scraping," Universitat Oberte de Catalunya, 2019.

[11] N. Gonzalo Soto, "Desarrollo de una API para datos abiertos," Universidad de La Laguna, 2018.

[12] M. F. Vargas Pulliquitín, "Minería de texto de la web, de opinión pública y hechos referentes al barrio la Floresta," Escuela Politécnica Nacional, 2018.

[13] M. Latorre, "Historia De Las Web," 2018.

[14] G. A. Falloux Acosta, "Diseño y desarrollo de un módulo de clasificación de páginas web en base a las características de su contenido utilizando técnicas de minería de datos," 2016.

[15] J. Lopez, “Web Scraping,” 2018.

[16] S. Patni, “Introduction - XML, JSON,” in *Pro RESTful APIs*, S. Patni, Ed. Apress, 2017, pp. 33–48.

[17] D. Chappell and T. Jewell, “Java Web Services,” 2002. [Online]. Available: https://books.google.fr/books?id=wiXOyXdvHO8C&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false. [Accessed: 23-Feb-2020].

[18] J. Sandoval, *RESTful Java Web Services*. Pack Publishing, 2009.

[19] J. Perriam, A. Birkbak, and A. Freeman, “Digital methods in a post-API environment,” *Int. J. Soc. Res. Methodol.*, vol. 00, no. 00, pp. 1–15, 2019.

[20] Q. T. Le and D. Pishva, “Application of Web Scraping and Google API Service to Optimize Convenience Stores’ Distribution,” pp. 2–6, 2015.

[21] D. A. Fernández Pérez, “Integración en ROS de herramientas de Web Scraping para análisis de vídeo,” Universidad de La Laguna, 2019.

[22] S. C, A. R, M. D. S, Suhendar, D. W, and R. M. A, “Web Scraping and Naïve Bayes Classification for Job Search Engine,” *iopscience.iop.org*, pp. 1–7, 2018.

[23] L. C. Dewi, Meiliana, and A. Chandra, “Social media web scraping using social media developers API and regex,” *Procedia Comput. Sci.*, vol. 157, pp. 444–449, 2019.

[24] U. J. Villanueva Rodríguez, “Investigación y Desarrollo de Técnicas de Scraping,” Universidad de Alcalá Escuela Politécnica Superior, 2019.

[25] D. López Angulo, “Euskal crawler,” 2016.

[26] Mozilla, “HTML.” [Online]. Available: https://developer.mozilla.org/es/docs/Web/HTML. [Accessed: 16-Feb-2020].

[27] w3schools, “Introduction to HTML.” [Online]. Available: https://www.w3schools.com/html/html_intro.asp. [Accessed: 16-Feb-2020].

[28] Mozilla, “HTTP.” [Online]. Available: https://developer.mozilla.org/es/docs/Web/HTTP. [Accessed: 16-Feb-2020].

[29] M. E. Raffino, “HTTP,” 29-Nov-2019. [Online]. Available: https://concepto.de/http/. [Accessed: 16-Feb-2020].

[30] LibCurl, “herramienta de línea de comandos y biblioteca para transferir datos con URL,” 2016. [Online]. Available: https://curl.haxx.se/. [Accessed: 13-Jan-2020].

[31] BeautifulSoup, “Biblioteca de Python para extraer datos de archivos HTML y XML,” 2012. [Online]. Available: https://www.crummy.com/software/BeautifulSoup/bs4/doc/. [Accessed: 13-Jan-2020].

[32] Unipython.com, “Solicitudes HTTP en Python con Requests.” [Online]. Available: https://unipython.com/solicitudes-http-en-python-con-requests/. [Accessed: 24-Aug-2020].

[33] riptutorial.com, “Python-requests.” [Online]. Available:

https://riptutorial.com/es/python-requests. [Accessed: 24-Aug-2020].

[34] Jsoup, "Analizador de HTML Java," 2017. [Online]. Available: https://jsoup.org/. [Accessed: 13-Jan-2020].

[35] Scrapy, "Un marco rápido y potente de raspado y rastreo web," 2007. [Online]. Available: https://scrapy.org/. [Accessed: 13-Jan-2020].

[36] J. Gonzalez, "Web Scraping con Scrapy Framework y Jupyter," 28-Nov-2019. [Online]. Available: https://josefgonzalez.me/es/post/scrapy-en-jupyter/. [Accessed: 14-Jun-2020].

[37] Web-Harvest, "herramienta de extracción de datos web de código abierto," 2006. [Online]. Available: http://web-harvest.sourceforge.net/. [Accessed: 13-Jan-2020].

[38] D. Glez-Peña, A. Lourenço, H. López-Fernández, M. Reboiro-Jato, and F. Fdez-Riverola, "Web scraping technologies in an API world," *Brief. Bioinform.*, vol. 15, no. 5, pp. 788–797, 2013.

[39] Api.github.com, "Documentación API GitHub." [Online]. Available: https://api.github.com/. [Accessed: 11-Sep-2020].

[40] Apify, "Raspado web, extracción de datos y automatización," 2019. [Online]. Available: https://apify.com/. [Accessed: 13-Jan-2020].

[41] Dexi.io, "Suite de inteligencia de comercio," 2019. [Online]. Available: https://dexi.io/. [Accessed: 13-Jan-2020].

[42] Diffbot, "Gráfico de conocimiento, AI Extracción de datos web y rastreo," 2018. [Online]. Available: https://www.diffbot.com/. [Accessed: 13-Jan-2020].

[43] Hunter.io, "Encuentra direcciones de correo electrónico en segundos ," 2017. [Online]. Available: https://hunter.io/. [Accessed: 13-Jan-2020].

[44] Import.io, "Extracción de datos, datos web, recolección web, preparación de datos, integración de datos," 2019. [Online]. Available: https://www.import.io/. [Accessed: 13-Jan-2020].

[45] Mozenda, "Software y servicios escalables de eliminación de datos web," 2015. [Online]. Available: https://www.mozenda.com/. [Accessed: 13-Jan-2020].

[46] Octoparse, "Herramienta de raspado web y rastreadores web gratuitos ," 2019. [Online]. Available: https://www.octoparse.com/. [Accessed: 13-Jan-2020].

[47] Parsehub, "Raspado web gratuito: el raspador web más potente," 2017. [Online]. Available: https://www.parsehub.com/. [Accessed: 13-Jan-2020].

[48] ScraperApi, "API Proxy para Scraping Web," 2019. [Online]. Available: https://www.scraperapi.com/. [Accessed: 13-Jan-2020].

[49] Webhose.io, "Convertimos contenido web no estructurado en fuentes de datos legibles por máquina," 2014. [Online]. Available: https://webhose.io/. [Accessed: 13-Jan-2020].

[50] 80legs.com, "Raspado web personalizable." [Online]. Available: http://80legs.com/. [Accessed: 13-Jan-2020].

[51] Java, "¿Qué es Java y para qué es necesario?," 2014. [Online]. Available: https://www.java.com/es/download/faq/whatis_java.xml. [Accessed: 12-Feb-2020].

[52] P. Corcuera, "Introducción a la Tecnología Java," 2017.

[53] E. A. Bocangel Lamas, "Java y Base de Datos," Moquegua, 1, 2016.

[54] J. C. García Monsálvez, "Python como primer lenguaje de programación textual en la Enseñanza Secundaria," *Redalyc*, vol. 18, no. 2, pp. 147–162, 2017.

[55] Python.org, "Python." [Online]. Available: https://www.python.org/. [Accessed: 11-Feb-2020].

[56] J. R. Molina Ríos, N. M. Loja Mora, M. P. Zea Ordóñez, and E. L. Loaiza Sojos, "Evaluación de los Frameworks en el Desarrollo de Aplicaciones Web con Python," *Rev. Latinoam. Ing. Softw.*, vol. 4, no. 4, pp. 1–7, 2016.

[57] P. R. Sangopanta Cajas, B. Mérelo Gil, Antonio, and E. E. Quinatoa Arequipa, "Analizar un código a través de lenguajes de programación C ++ y Code :: Blocks," *Ciencias la Ing. y Apl.*, vol. 3, pp. 37–53, 2019.

[58] Albatros, "A Brief Description C ++." [Online]. Available: http://www.cplusplus.com/info/description/. [Accessed: 15-Feb-2020].

[59] Semalt, "¿Cuáles son los mejores lenguajes de programación para raspar un sitio?" [Online]. Available: https://semalt.com/es/qa/5451-datos-de-la-pagina-web.htm. [Accessed: 15-Feb-2020].

[60] PHP, "¿Qué es PHP?," 10-Jun-2009. [Online]. Available: https://www.php.net/manual/es/intro-whatis.php. [Accessed: 15-Feb-2020].

[61] L. R. Julian and F. Natalia, "The use of web scraping in computer parts and assembly price comparison," *CONMEDIA 2015 - Int. Conf. New Media 2015*, 2016.

[62] NubeColectiva, "Que es PHP y otros detalles," 11-Nov-2018. [Online]. Available: https://blog.nubecolectiva.com/que-es-php-y-otros-detalles/. [Accessed: 15-Feb-2020].

[63] N. Caballero Sánchez, "Implementación en lenguaje Perl de algoritmos de recuento de k -meros y su aplicación al ensamblaje de novo de genomas," 2017.

[64] J. D. Luján, "Web scraping," 2017. [Online]. Available: https://ed.team/blog/web-scraping. [Accessed: 15-Feb-2020].

[65] E. A. Lozano Sánchez, "Algoritmo de Búsqueda y Recomendación para Reparar Enlaces en Páginas Web," 2017.

[66] Django, "Django." [Online]. Available: https://www.djangoproject.com/. [Accessed: 24-Aug-2020].

[67] EspiFreelancer, "Que es el patrón MTV (Model Template View)." [Online]. Available: https://espifreelancer.com/mtv-django.html. [Accessed: 24-Aug-2020].

[68] F. J. Lopez-Pellicer, R. Béjar, M. A. Latre, J. Nogueras-Iso, and F. J. Zarazaga-Soria, "GitHub como herramienta docente," *Actas las XXI Jornadas la Enseñanza Univ. la*

Informática, pp. 66–73, 2015.

[69] GitHub, "Plataforma de desarrollo inspirada en tu forma de trabajar," 2008. [Online]. Available: https://github.com/. [Accessed: 13-Jan-2020].

[70] Stack Overflow, "Stack Overflow Annual Developer Survey," 15-Feb-2020. [Online]. Available: https://insights.stackoverflow.com/survey. [Accessed: 21-Jun-2020].

[71] A. Echezarraga Porto, "Sistema de detección de intrusos mediante cámara," 2016.

[72] J. De León Santana, "Desarrollo de una aplicación que facilita la enseñanza y el estudio del póker en su variante Texas Hold ' em."

[73] Bitbucket, "La solución Git para equipos profesionales," 2019. [Online]. Available: https://bitbucket.org/product/. [Accessed: 13-Jan-2020].

[74] J. Dompablo Tobar, "DevOps para automatización de Gitlab en alta disponibilidad," Universidad Autónoma de Madrid Escuela Politécnica Superior, 2018.

[75] GitLab, "Descubra por qué GitLab fue evaluado como Líder por una firma de investigación independiente," 2014. [Online]. Available: https://about.gitlab.com/. [Accessed: 13-Jan-2020].

[76] M. Rouse, "¿Qué es Autenticación multifactor (MFA)?," Jun-2014. [Online]. Available: https://searchdatacenter.techtarget.com/es/definicion/Autenticacion-multifactor-MFA. [Accessed: 15-Feb-2020].

[77] IONOS, "Alternativas a GitHub: las 5 mejores aplicaciones," 06-Aug-2019. [Online]. Available: https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/alternativas-a-github/. [Accessed: 15-Feb-2020].

[78] GitKraken, "GitKraken." [Online]. Available: https://www.gitkraken.com/. [Accessed: 15-Feb-2020].

[79] ApacheAllura, "Apache Allura." [Online]. Available: https://allura.apache.org/. [Accessed: 15-Feb-2020].

[80] S. Plaza Estévez, N. Ramírez Lamela, and C. Acosta Morales, "API de servicios web orientados a accesibilidad," 2016.

[81] GitHub, "GitHub API v3 | GitHub Developer Guide," 2011. [Online]. Available: https://developer.github.com/v3/. [Accessed: 17-Feb-2020].

[82] I. Sánchez Prieto, "Herramienta de análisis automático de vulnerabilidades SSL," Universidad Autónoma de Madrid Escuela Politécnica Superior, 2016.

[83] E. A. Oviedo Carrascal, A. I. Oviedo Carrascal, and G. L. Vélez Saldarriaga, "Minería de datos: Aportes y tendencias en el servicio de salud de ciudades inteligentes," *Rev. Politécnica*, vol. 11, no. 1, pp. 111–120, 2019.

[84] C. A. Córdova Sáenz, "Metodología basada en Minería de Datos para la detección de usuarios Influencers en Twitter," Universidad Nacional de Trujillo, 2019.

[85] K. P. Kalyanathaya, D. Akila, and P. Rajesh, "Advances in natural language processing

–a survey of current research trends, development tools and industry applications," *Int. J. Recent Technol. Eng.*, vol. 7, no. 5, pp. 199–201, 2019.

[86] E. Gil, *Big data, privacidad y protección de datos*, no. June 2016. Madrid, España, 2016.

[87] M. Bayat, M. Khanzadi, F. Nasirzadeh, and A. Chavoshian, "Financial conflict resolution model in BOT contracts using bargaining game theory," *Constr. Innov.*, vol. 20, no. 1, pp. 18–42, 2020.

[88] M. Casillo, F. Clarizia, G. D'Aniello, M. De Santo, M. Lombardi, and D. Santaniello, "CHAT-Bot: A cultural heritage aware teller-bot for supporting touristic experiences," *Pattern Recognit. Lett.*, vol. 131, pp. 234–243, 2020.

[89] C. E. Higuera Quishpe, "Creación de un chatbot con naturtal language process y Deep learning para la interacción humano-bot," 2019.

[90] J. I. Duarte F., G. E. Rodríguez G., J. Lares, and J. R. Sosa B., "Venezolanos en Twitter: Humanos, Bots o Ciborgs, Modelo de V Clasificación," *Rev. Ing.*

[91] J. Mera-Paz, "Análisis del proceso de pruebas de calidad de software," *Ing. Solidar.*, vol. 12, no. 20, pp. 163–176, 2016.

ANEXOS

ANEXO 1

SCRIPT PYTHON WEB SCRAPING – API GITHUB

```
repositorios_git = []

# Documentacion del API: https://api.github.com/
endpoint = "https://api.github.com/users/Josheriff/repos?"
#"https://api.github.com/users/Josheriff/repos?"


response = requests.get(endpoint)

# RESPUESTA ESTA EN FORMATO JSON
repositorios = response.json() # puedo utilizar la libreria json para
ver de la mejor manera
for repositorio in repositorios:
    #print(json.dumps(response.json(), indent=4))
    itulo = repositorio["name"].replace('-', ' ').replace('/',
'').replace('_', ' ').strip()
    descripcion = repositorio["description"]  # .replace('-',
'').replace('/', '').replace('_', ' ').strip()
    star = repositorio["stargazers_count"]
    fork = repositorio["forks_count"]
    fecha_creacion = repositorio["created_at"]
    fecha_actualizacion = repositorio["updated_at"]
    url_commits = repositorio["commits_url"].replace('{/sha}',
'').strip()
    url_colaboradores = repositorio["contributors_url"]
    url_lenguajes = repositorio["languages_url"]
    url_repositorio = repositorio["svn_url"]

    col.update_one(
        {
            "titulo": titulo
            # "descripcion": descripcion
        }, {
            "$set": {
                "descripcion": descripcion,
                "star": star,
                "fork": fork,
                "fecha_creacion": fecha_creacion,
                "fecha_actualizacion": fecha_actualizacion,
                "url_commits": url_commits,
                "url_colaboradores": url_colaboradores,
                "url_lenguajes": url_lenguajes,
                "url_repositorio": url_repositorio,
                "titulo": titulo
            }
        }, upsert=True)
```

ANEXO 2

SCRIPT DE ADMINISTRADOR DE DJANGO – CLASES DE PYTHON

```
class RepositorioAdmin(admin.ModelAdmin):
   list_display =
('titulo','descripcion','star','fork','fecha_creacion','fecha_actualiz
```

```
acion',

'url_commits','url_colaboradores','url_lenguajes','url_repositorio')


class UserAdmin(admin.ModelAdmin):
   list_display = ('nombre','apellido','email','password')
   search_fields = ('nombre','apellido')
```

ANEXO 3

SCRIPT FRONT-END – PYTHON-HTLM

```
{% block main %}
    <link href="{% static 'css/index.css' %}" rel="stylesheet" />
   <!--<br>
   <a class="btn btn-primary" href="{% url 'repositorios:create_user'
%}">Nuevo Usuario</a>-->
   <p align="center" class="text-success fa-2x">Repositorios
Scrapeadas</p>
   <div>
      <table class="table-details-git"  border="1">
         <thead>
            <tr>
                  <th>Título</th>
                  <th>Descripción</th>
               <!--<th># Watch</th>-->
               <th>Star</th>
               <th>Fork</th>
               <th>Fecha Creación</th>
               <th>Fecha Actualización</th>
               <th>Commits</th>
               <th>Colaboradores</th>
                   <th>Lenguajes</th>
                   <th>URL Repositorio</th>
               </tr>
            </thead>
            <tbody>
            {% for item in repositorio %}
                  <tr>
                  <td>{{ item.titulo }}</td>
                  <td>{{ item.descripcion }}</td>
                  <td>{{ item.star }}</td>
                  <td>{{ item.fork }}</td>
                  <td>{{ item.fecha_creacion }}</td>
                  <td>{{ item.fecha_actualizacion }}</td>
                  <!--<td><a href="{{ item.url_commits }}"
target="_blank">Ver Commits</a></td>-->
                  <td><a href="{{ item.url_commits }}"
target="_blank"><center>Ver Commits</center></a></td>

                  <td><a href="{{ item.url_colaboradores }}"
target="_blank"><center>Ver Colaboradores</center></a></td>
                  <td><a href="{{ item.url_lenguajes }}"
target="_blank"><center>Ver Lenguajes</center></a></td>
                  <td><a href="{{ item.url_repositorio }}"
target="_blank"><center>Ver Repositorio</center></a></td>

               </tr>
```

```
            {% endfor %}
          </tbody>
        </table>
    </div>
{% endblock %}
```

EN CASO DE REQUERIR EL SCRIPT DE EJECUCIÓN COMPLETO, COMUNICARSE CON EL AUTOR DE ESTE DOCUMENTO.