

**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE QUITO**

**CARRERA:  
INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:  
Ingeniero e Ingeniera de Sistemas**

**TEMA:  
CREACIÓN DE UNA APLICACIÓN WEB PARA LA EXTRACCIÓN DE DATOS  
DE PÁGINAS WEB ACERCA DE PROCESOS DE LA MATRICULACIÓN  
VEHICULAR UTILIZANDO LA TÉCNICA DE WEB SCRAPING.**

**AUTOR Y AUTORA:  
BYRON VLADIMIR MONTÚFAR CHIQUITO  
IVONNE PAULINA MULLO RODRÍGUEZ**

**TUTOR:  
FRANKLIN EDMUNDO HURTADO LARREA**

**Quito, marzo del 2021**

## CESIÓN DE DERECHOS DE AUTOR

Nosotros, BYRON VLADIMIR MONTÚFAR CHIQUITO, con documento de identificación N° 1720526837, e IVONNE PAULINA MULLO RODRÍGUEZ con documento de identificación N° 1721586301, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación con el tema: CREACIÓN DE UNA APLICACIÓN WEB PARA LA EXTRACCIÓN DE DATOS DE PÁGINAS WEB ACERCA DE PROCESOS DE LA MATRICULACIÓN VEHICULAR UTILIZANDO LA TÉCNICA DE WEB SCRAPING, mismo que ha sido desarrollado para optar por el título de INGENIEROS DE SISTEMAS en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

-----  
**Byron Vladimir  
Montúfar Chiquito**  
**CI: 1720526837**

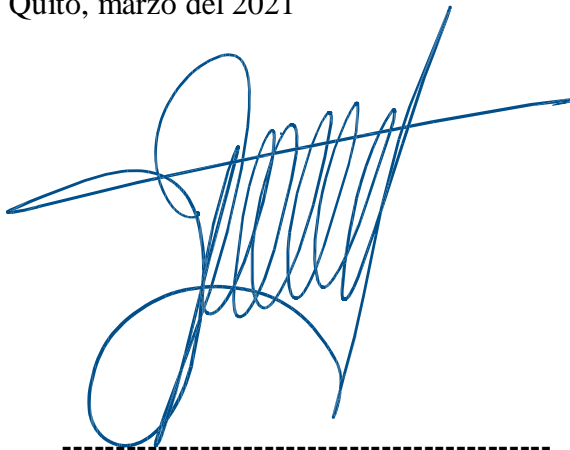
-----  
**Ivonne Paulina  
Mullo Rodríguez**  
**CI: 1721586301**

Quito, marzo del 2021

## DECLARACIÓN DE COAUTORÍA DEL DOCENTE TUTOR

Yo, FRANKLIN EDMUNDO HURTADO LARREA, declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, con el tema: CREACIÓN DE UNA APLICACIÓN WEB PARA LA EXTRACCIÓN DE DATOS DE PÁGINAS WEB ACERCA DE PROCESOS DE LA MATRICULACIÓN VEHICULAR UTILIZANDO LA TÉCNICA DE WEB SCRAPING, realizado por BYRON VLADIMIR MONTÚFAR CHIQUITO e IVONNE PAULINA MULLO RODRÍGUEZ, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerado como trabajo final de titulación.

Quito, marzo del 2021



-----  
**Franklin Edmundo Hurtado Larrea**  
**CI: 1713382016**

## **Dedicatoria**

Quiero iniciar dedicando este proyecto a mi abuelita “Mamia” que siempre me brindo todo su apoyo y trabajó muy fuerte a pesar de que ya estaba viejita para darme lo necesario y poder obtener mi título en esta prestigiosa Universidad.

Siempre me impulso a seguir adelante y no decaer a luchar por mis objetivos y nunca rendirme, vi en ella su fortaleza, su dedicación para ir cada día a trabajar para que yo pueda seguir estudiando.

A mi madre “La mami Pato” que a pesar de que no vivía con ella siempre me visitaba y se sentía orgullosa que siguiera estudiando y dando esa palabra de aliento que me impulsaba a cumplir el objetivo.

A mi esposa que siempre me animaba a trabajar y estudiar cuando ya sentía desánimo y la frustración que se generaba en esta difícil etapa.

Tal vez este documento no exprese todo lo que siento, pero siempre estaré agradecido con estas tres mujeres que me llevaron por el buen camino. A ellas gracias por estar ahí, día tras día, sin ustedes no podría haber llegado este momento.

También quiero dedicar esto a mi hijo “Didy” que cuando lo lea sepa que siempre lo apoyaré en su camino y así como lo hicieron conmigo tendrá los consejos, palabras de aliento, y todo lo que pueda darle, para que pueda cumplir sus metas.

**Byron Vladimir Montúfar Chiquito**

## **Dedicatoria**

A mi madre Inés Rodríguez por su amor, dedicación y porque a través de su esfuerzo me ha permitido alcanzar esta meta. Gracias por creer en mí, por apoyarme siempre y nunca dejarme caer. Me llenas de mucho orgullo madre mía ya que a pesar de las dificultades que tenías siempre encontraste una manera para sacarme adelante, no va a haber manera de devolverte tanto. Este proyecto es un logro más que llevo a cabo y sin lugar a duda ha sido en gran parte gracias a ti, no sé en donde me encontraría de no ser por tu ayuda, tu compañía y tu amor.

A mi bebe Didier Montúfar M. aunque estas muy pequeñito, tú eres lo más importante que tiene papi y mami en este mundo, eres nuestra razón para esforzarnos cada día y nos motivas a seguir adelante. Gracias por llenar nuestras vidas de felicidad.

Y finalmente quiero agradecer a mi esposo Byron Montúfar por estar siempre a mi lado en todos los momentos primero como amigos, enamorados y ahora como esposos. Gracias por brindarme tu apoyo y amor cada día.

**Ivonne Paulina Mullo Rodríguez**

## **Agradecimientos**

A las personas que siempre nos dieron su apoyo incondicional, brindándonos una guía en toda esta ardua carrera que hoy culmina con éxito.

A nuestro tutor de proyecto Ingeniero Franklin Hurtado, por darnos su tiempo y apoyo en la realización del presente trabajo exigiéndonos investigación y análisis para terminar con satisfacción esta carrera.

A la Universidad por acogernos y enseñarnos a ser profesionales de éxito con los más altos valores morales en cualquier circunstancia que se presente en nuestro camino profesional.

**Byron Vladimir Montúfar Chiquito**  
**Ivonne Paulina Mullo Rodríguez**

# ÍNDICE

INTRODUCCIÓN .....	1
Problema de Estudio .....	1
Justificación.....	2
Objetivo General.....	3
Objetivos Específicos.....	3
Marco Metodológico.....	3
Análisis de Riesgos.....	6
CAPÍTULO 1 .....	8
Marco referencial y teórico. ....	8
1.1 Marco Referencial.....	8
1.1.1 Agencia Nacional de Tránsito (ANT) .....	8
1.1.2 Agencia Metropolitana de Tránsito (AMT).....	8
1.1.3 Autoridad de Tránsito Municipal (ATM).....	9
1.1.4 Empresa Pública Municipal de Movilidad, Tránsito y Transporte de Cuenca (EMOV) .....	10
1.2 Marco Teórico.....	10
1.2.1 Servidor Web .....	10
1.2.2 Sitio Web .....	10
1.2.3 Certificado Digital.....	10
1.2.4 Web Scraping.....	11
1.2.5 Bases de datos no relacionales.....	12
1.2.6 MongoDB .....	13
1.2.7 Apache.....	13
1.2.8 ChatBot.....	13
1.2.9 JSON .....	13
1.2.10 OctoPerf.....	14
1.2.11 OWASP Zap .....	14
1.2.12 Lenguaje de programación Python.....	14
1.2.12.1 Características Python .....	14
1.2.12.2 Ventajas y Desventajas .....	15
1.2.13 Arquitectura MVC.....	16
1.2.13.1 Modelos .....	16
1.2.13.2 Vistas .....	16
1.2.13.3 Controladores .....	16
1.2.14 Gestión de la calidad del software .....	17
1.2.15 Pruebas de Software .....	17

1.2.15.1 Métricas .....	17
1.2.16 Estándar ISO 9126 .....	18
1.2.16.1 Características Propuestas por ISO 9126.....	18
1.2.17 Metodología XP (eXtreme Programming) .....	19
1.2.17.1 Características fundamentales de XP .....	19
1.2.17.2 Prácticas XP .....	20
1.2.17.3 Modelado Clase-Responsabilidad-Colaborador (CRC) .....	20
1.2.18 Adaptación de la Metodología XP .....	20
1.2.19 Trazabilidad Ágil .....	21
1.2.19.1 Matriz de Trazabilidad.....	22
<b>CAPÍTULO 2 .....</b>	<b>23</b>
Análisis y Diseño .....	23
2.1. Requerimientos Funcionales y No Funcionales.....	23
2.1.1 Análisis de la estructura de las páginas AMT, ATM, EMOV .....	25
2.1.1.1 Agencia Metropolitana de Tránsito (AMT).....	26
2.1.1.2 Autoridad de Tránsito Municipal (ATM).....	28
2.1.1.3 Empresa Pública Municipal de Movilidad, Tránsito y Transporte de Cuenca (EMOV) .....	29
2.1.2 Diagrama de Caso de Uso de la aplicación Web .....	30
2.1.3 Actor.....	31
2.1.4 Diagrama de actividades del sistema .....	32
2.1.5 Arquitectura lógica del sistema.....	34
2.1.6 Modelado Clase-Responsabilidad-Colaborador (CRC).....	35
2.1.7 Diagrama de clases.....	36
2.1.8 Estructura de la base de datos no relacional .....	37
2.1.8.1 Esquema conceptual del archivo JSON.....	37
<b>CAPITULO 3 .....</b>	<b>39</b>
Construcción y Pruebas.....	39
3.1 Estándares de programación de Python .....	39
3.2 Construcción de la base de datos no relacional .....	42
3.3 Métodos .....	46
3.4 Proceso de extracción de datos de las páginas AMT, ATM, EMOV .....	49
3.5 Código Relevante.....	50
3.5.1 Codificación Algoritmo de Web Scraping .....	50
3.6 Pruebas .....	52
3.6.1 Plan de Pruebas .....	52
3.6.1.1 Objetivos de calidad de software .....	52



3.6.2 Pruebas de exactitud.....	53
3.6.3 Pruebas de rendimiento .....	61
3.6.4 Pruebas de seguridad.....	64
3.6.5 Pruebas de usabilidad .....	67
3.7 Perspectiva visual del sistema .....	68
3.8 Limitaciones y Restricciones.....	74
CONCLUSIONES .....	76
RECOMENDACIONES .....	78
Referencias .....	79

## ÍNDICE DE TABLAS

Tabla 1: Ventajas y desventajas del Web Scraping .....	11
Tabla 2: Requerimientos del sistema .....	24
Tabla 3: Actor del sistema.....	31
Tabla 4: Métodos aplicación web .....	46
Tabla 5: Caso de uso - Inicio de sesión.....	53
Tabla 6: Caso de uso - Generación JSON con URL específica .....	54
Tabla 7: Caso de Uso - Generación JSON con dominio principal .....	54
Tabla 8: Caso de uso - Generación de Reportes.....	55
Tabla 9: Caso de uso - Consumo de datos mediante servicio web.....	56
Tabla 10: Caso de uso - Historial de resultados .....	56
Tabla 11: Caso de prueba Nro. 1 .....	57
Tabla 12: Caso de prueba Nro.2 .....	58
Tabla 13: Caso de prueba Nro.3 .....	58
Tabla 14: Caso de prueba Nro.4.....	59
Tabla 15: Caso de prueba Nro.5 .....	59
Tabla 16: Caso de prueba Nro.6.....	60
Tabla 17: Resultados desarrollo casos prueba.....	60
Tabla 18: Descripción del nivel de alarma de seguridad .....	65
Tabla 19: Medición de eficacia .....	68

## ÍNDICE DE FIGURAS

Figura 1: Certificado de seguridad - Disculpas públicas de la ATM.....	9
Figura 2: Preguntas Centrales de acuerdo con la característica de calidad .....	18
Figura 3: Proceso XP para el desarrollo de software.....	19
Figura 4: Sección de Servicios de la página oficial de la AMT .....	26
Figura 5: Divisiones de la información del servicio de la AMT .....	27
Figura 6: Sección de Servicios de la página oficial de la AMT .....	28
Figura 7: Divisiones de la información del servicio de la ATM .....	29
Figura 8: Sección de Servicios de la página oficial de la EMOV .....	29
Figura 9: Divisiones de la información del servicio de la EMOV .....	30
Figura 10: Diagrama de caso de uso del sistema.....	31
Figura 11: Diagrama de actividades del sistema .....	33
Figura 12: Arquitectura lógica del sistema .....	34
Figura 13: Diagrama de Clases.....	36
Figura 14: Ejemplo de indentación.....	39
Figura 15: Separación de importaciones .....	40
Figura 16: Separación de operadores binarios .....	40
Figura 17: Ejemplo de comentarios .....	41
Figura 18: Ejemplo de nombramiento .....	41
Figura 19: Ejemplo de nombres de las clases.....	41
Figura 20: Líneas de código pertenecientes a webscrap.py. ....	42
Figura 21: Estado del servicio de la base de datos .....	42
Figura 22: Levantamiento el servicio de MongoDB .....	43
Figura 23: Estado del servicio de la base de datos .....	43
Figura 24: Importación de librería componentes MongoDB .....	43
Figura 25: Ejecución del comando mongo.....	44
Figura 26: Ejecución del comando use bdd_webscrapping .....	44
Figura 27: Ejecución del comando show collections.....	44
Figura 28: Proceso de enlistar información.....	45
Figura 29: Codificación Algoritmo de Web Scraping 1 .....	51
Figura 30: Codificación Algoritmo de Web Scraping 2 .....	51
Figura 31: Codificación Algoritmo de Web Scraping 3 .....	52
Figura 32: Captura de pantalla de Octoperf en ejecución .....	62
Figura 33: Sección de sumario de Octoperf .....	62
Figura 34: Sección HITS de Octoperf .....	63
Figura 35: Sección de latencia de Octoperf .....	63
Figura 36: Sección de Top Chart de Octoperf.....	64
Figura 37: Owasp Zap en ejecución .....	66
Figura 38: Reporte de resultados de escaneo de Owasp Zap .....	66
Figura 39: Detalle de alertas Owasp Zap .....	67
Figura 40: Interfaz para el inicio de sesión en el sistema .....	69
Figura 41: Interfaz principal del usuario Administrador .....	70
Figura 42: Opción del menú principal: webscrapping .....	70
Figura 43: Ejecución del webscrapin por url del servicio .....	71
Figura 44: Resultado del webscraping por URL del servicio .....	72
Figura 45: Proceso de Web Scraping por url principal.....	72
Figura 46: Sección Históricos del sistema .....	73
Figura 47: Archivo JSON resultante del proceso de Web Scraping.....	73
Figura 48: Archivo JSON descargado del Web Scraping .....	74

## **Resumen**

En el siguiente trabajo de titulación se ha planteado y logrado el desarrollo de una aplicación web para la extracción de datos acerca de procesos de la matriculación vehicular de las páginas oficiales de las instituciones con competencias de tránsito ubicadas en Quito, Guayaquil y Cuenca, utilizando la técnica de Web Scraping. La información extraída se almacena en un archivo JSON y se estructura para que sirva como entrada de datos al ChatBot que se desarrolló en paralelo como proyecto de titulación en la carrera de Ingeniería de Sistemas de la Universidad Politécnica Salesiana y cuyo objetivo es ayudar a los usuarios acceder de una manera más sencilla y de forma precisa a la información acerca del proceso de matriculación vehicular en las ciudades antes mencionadas. A lo largo del documento se describen los aspectos fundamentales del proceso de desarrollo y del sistema como, los objetivos generales y específicos, la metodología ágil Extreme Programming (XP), los requerimientos funcionales y no funcionales, la codificación de algoritmos en lenguaje Python, las interfaces de usuario, así como las pruebas que se aplican para asegurar la calidad del software.

**Palabras Clave:** Web Scraping, Algoritmo, ChatBot, JSON

## **Abstract**

In the following degree work, the development of a web application has been proposed and achieved for the extraction of data about the processes of vehicle registration from the official pages of the institutions with traffic powers located in Quito, Guayaquil and Cuenca, using the Web Scraping technique. The information extracted is stored in a JSON file and is structured to serve as data input to the ChatBot that was developed in parallel as a degree project in the Systems Engineering career of the Salesian Polytechnic University and whose objective is to help users access in a simpler and more precise way the information about the vehicle registration process in the aforementioned cities. Throughout the document the fundamental aspects of the development process and the system are described, such as the general and specific objectives, the agile Extreme Programming (XP) methodology, the functional and non-functional requirements, the coding of algorithms in Python language, the user interfaces, as well as the tests that are applied to ensure the quality of the software.

**Keywords:** Web Scraping, Algorithm, ChatBot, JSON

# **Introducción**

## **Problema de Estudio**

Dado que el Internet es una red mundial de computadoras, donde los usuarios pueden acceder a la información y tener inclusive comunicación directa con otros usuarios desde cualquier parte del mundo y a cualquier hora; se ha convertido en un medio de difusión de información extremadamente eficiente y es por ello por lo que actualmente existe una enorme cantidad de sitios web, donde los usuarios pueden realizar búsquedas de información según sus necesidades. En la actualidad la cantidad de sitios web ha crecido considerablemente, en la mayoría de los casos, los usuarios deben consultar varios de ellos para poder localizar la información que requieren esto implica que deben contar con un nivel de alfabetización digital adecuado.

De acuerdo con las cifras publicadas por el (Instituto Nacional de Estadística y Censos [INEC], 2020) actualmente la población en Ecuador asciende a más de 17.450.000 habitantes. (INEC, 2017) “El 10,5% de las personas de 15 a 49 años en el Ecuador son analfabetas digitales.” Apenas el 18,9% de las personas de entre 55 y 64 años ha usado una computadora. (p. 38) Por otro lado en (Diario El Comercio, 2020) fue publicado que: el director de Registro y Administración Vehicular de la AMT, informó que en el 2019 se matricularon 465.900 carros en la capital y quedaron rezagados aproximadamente 35.000.” (p. 7)

Realizando una extrapolación de las cifras obtenidas de dichas fuentes, Se concluye que de los más de 465.000 usuarios que realizaron el proceso de matriculación de su vehículo el año pasado un 10% de ellos no posee un buen nivel de alfabetización digital, lo que se resume en la necesidad de facilitar el acceso a la información correspondiente a dicho proceso a través de un canal diferente, que reduzca el tiempo y esfuerzo que emplea el usuario para esta tarea.

Frente a esta problemática, en la carrera de Ingeniería de Sistemas de la Universidad Politécnica Salesiana se planteó un proyecto de titulación que busca el desarrollo de un ChatBot que permita a los usuarios acceder de una manera más sencilla y de forma precisa a la información acerca del proceso de matriculación vehicular en Quito, Guayaquil y Cuenca. Actualmente, la información existente en las páginas oficiales de la Agencia Metropolitana de Tránsito de dichas ciudades y que nutrirá al ChatBot es obtenida de forma manual, lo cual consume mucho tiempo, esfuerzo y está sujeta al error humano (confusiones, omisiones, alteraciones involuntarias, etc), como parte de este proyecto es necesario el desarrollo de un componente automatizado que proporcione los datos de entrada a este ChatBot de forma rápida, oportuna y estructurada.

### **Justificación**

Una vez analizada la problemática anteriormente planteada se logra determinar que, para acelerar el proceso de extracción de datos, disminuir el riesgo de error humano, mantenerlos actualizados, estructurarlos y usarlos como entrada en el ChatBot es conveniente hacer uso de una técnica web llamada Web Scraping, la cual consiste en utilizar herramientas de software para extraer información específica de diferentes sitios web de manera automatizada.

Esto permitirá a los desarrolladores del ChatBot reducir en gran medida el tiempo y el esfuerzo que implica recabar la información correspondiente al proceso de matriculación vehicular de los sitios web oficiales de las agencias metropolitanas de tránsito en cada una de las 3 principales ciudades (Quito, Guayaquil y Cuenca), estructurarla y transformarla en datos de entrada para el ChatBot.

## **Objetivo General**

Desarrollar una aplicación web que permita automatizar la extracción de la información de páginas web utilizando la técnica de Web Scraping para conocer los procesos de matriculación vehicular.

## **Objetivos Específicos**

Investigar la utilidad, ventajas y desventajas de la técnica Web Scraping que se usará para la extracción de datos en la web.

Analizar los requerimientos planteados por el ChatBot para estructurar los datos y para facilitar su futura integración con los resultados del Web Scraping.

Construir la aplicación web que permita la extracción automática de información digital correspondiente a los procesos de matriculación vehicular.

Realizar las pruebas unitarias y de funcionalidad que permitan verificar si se está cumpliendo con todos los requerimientos del sistema.

## **Marco Metodológico**

Las metodologías ágiles fueron concebidas para adaptar la dinámica de trabajo a las condiciones del proyecto, aumentando la flexibilidad en la respuesta para ajustar su desarrollo a las circunstancias específicas del entorno. (IEBS, 2019)

En este caso en particular, se utilizó la metodología ágil XP, ya que define las diferentes actividades de desarrollo del software de forma iterativa e incremental, ajustándose a los objetivos planteados. Pero, cuando en el proceso de software no se encuentra claramente definido el usuario final, o se carece de este, ¿Cómo pueden elaborarse las historias de usuario?

En un proyecto de software con esta característica el equipo de desarrolladores debe recurrir a la utilización de otra herramienta, en este caso, una matriz ágil de trazabilidad que les permita definir, priorizar y dar seguimiento al conjunto de requerimientos del



sistema, así como las iteraciones necesarias para obtener el producto final.  
(Sustentación teórica en la pág. 21)

En el desarrollo de este software se aplicaron todas las fases que comprenden a la metodología, dentro de las cuales se concentran las siguientes actividades:

Actividades N°1, se determinaron los requerimientos funcionales y no funcionales del sistema a través de una matriz de trazabilidad ágil (Anexo N° 01). Con la ayuda de la estructura del archivo JSON propia del ChatBot facilitada por el equipo de desarrollo del mismo, se analizó la estructura de cada una de las páginas de origen de la información (AMT, ATM y EMOV) logrando diseñar un esquema de presentación de la información de cada página que se almacenan en el archivo JSON resultante del Web Scraping. Posteriormente se comenzó con la programación de la sección de inicio de sesión del software. Adicionalmente se realizaron pruebas de funcionalidad para verificar que el módulo de inicio de sesión se ejecutaba correctamente. En esta iteración se obtuvieron como resultados: la matriz de trazabilidad, la estructura del archivo JSON y el primer módulo codificado del software (Inicio de sesión).

Actividades N°2, a través del análisis de la matriz de trazabilidad se determinaron los riesgos y se plantearon las cartillas de casos de pruebas de funcionamiento y así verificar y validar que la codificación del algoritmo cubra todos los requerimientos planteados, se identificó el único actor del sistema que es el Administrador, se diseñó un Diagrama de Caso de Uso (DCU) generalizado y el esquema de modelo arquitectónico basado en los componentes lógicos del sistema, utilizando lenguaje UML y una herramienta de diagramación online. Los resultados obtenidos son: el DCU, la matriz de análisis de riesgo (Anexo N° 02), las cartillas de casos de prueba, el DCU y el diseño arquitectónico del sistema.

Actividades N°3, se diseñó el diagrama de actividades del sistema para mostrar gráficamente las actividades y secuencias de estas dentro del sistema y las tarjetas CRC correspondientes a cada clase del sistema, como lo recomienda la metodología XP. Se comenzó el proceso de construcción de los algoritmos de Web Scraping recurriendo a la programación en pareja y se utilizó el lenguaje de programación Python debido a sus características y ventajas (más información en el marco teórico) así como los estándares de programación PEP con la finalidad de asegurar la calidad del código. Obteniendo como resultados: el diagrama de actividades del sistema, las tarjetas CRC y el algoritmo inicial de Web Scraping.

Actividades N°4, se construyó la base de datos no relacional de acuerdo con las características propias del archivo JSON resultante de la aplicación del Web Scraping. El algoritmo que se logró codificar está dividido en 8 pasos, se realizaron pruebas de funcionamiento del algoritmo para verificar que la extracción de datos se realizó correctamente y se validó la información obtenida mediante una comparación con la información mostrada en las páginas web de origen. Se detectaron algunos errores en la validación de la información extraída y se procedió a revisar y analizar nuevamente las actividades realizadas en las actividades N°1. Se modificó el algoritmo, eliminando secciones innecesarias, simplificando funciones y mejorando el código, logrando la aplicación de correcciones para optimizar el algoritmo de forma tal que permitiera la correcta extracción de los datos necesarios para alimentar al ChatBot. Los resultados obtenidos en esta iteración son: la base de datos donde se almacenan los archivos JSON y el algoritmo final (depurado) de Web Scraping.

Actividades N°5, se documentaron algunas secciones del algoritmo final en forma de diagrama de flujo según el patrón UML para una mayor comprensión del proceso de extracción de datos. Se tomaron como referencia el estándar ISO 9126 y las métricas

de calidad de software planteadas por Pressman (2010) y se plantearon los requerimientos de entorno y los objetivos de calidad que persiguen. Se seleccionó de entre todos los tipos de pruebas las que se pueden aplicar al sistema, de acuerdo con su naturaleza: pruebas de seguridad, de rendimiento, de usabilidad y de exactitud; haciendo uso de herramientas como: Octoperf para las pruebas de rendimiento (pruebas de carga) y OWASP ZAP para probar la seguridad del sistema, para el resto de las pruebas aplicadas se utilizó la matriz de trazabilidad y las cartillas de casos de pruebas diseñadas en las actividades N°2, y para las pruebas de usabilidad se usó la tabla de medición de eficacia. Finalmente se obtuvieron como resultados: el diagrama de flujo de algunas secciones del algoritmo de Web Scraping y la tabla de resumen de plan de pruebas (Anexo N° 08).

Cabe mencionar que a medida en que se fue desarrollando el software se aplicaron pruebas unitarias, se analizaron los resultados obtenidos en estas, se plasmaron en una tabla de Resumen de resultados del desarrollo de los casos de pruebas (p.62), finalmente se redactaron las conclusiones y recomendaciones en base a estos resultados los cuales fueron satisfactorios de acuerdo con los objetivos de calidad planteados.

### **Análisis de Riesgos**

En esta sección se realizó la recopilación, evaluación y registro de la información necesaria para plantear medidas en respuesta ante un peligro determinado, hecho fortuito o condición adversa que pueda suscitarse durante el proceso de desarrollo del software o en su implementación, tomando en cuenta que estos sucesos pueden acontecer sobre el software, sobre los actores, o sobre el equipo de desarrollo de este. Algunos de los riesgos que se han podido detectar durante el análisis de los requerimientos y que afectarían de forma negativa al software, así como los riesgos

que pueden afectar de igual manera al equipo de desarrollo están representados en el Anexo N° 02: Matriz de análisis de riesgos.

# Capítulo 1

## Marco referencial y teórico

### 1.1 Marco Referencial

Se realizó una búsqueda de fuentes teóricas que respalden la selección de las agencias de tránsito que serían objetivo del proceso de Web Scraping, logrando obtener información de fuentes como:

(Diario EL TELÉGRAFO, 2014) según la ANT desde el año 2014 por una disposición que consta en el artículo 130 del Código Orgánico de Ordenamiento Territorial y Descentralización (COOTAD) veinticinco de los doscientos veintidós municipios del territorio nacional manejarán competencias de tránsito, pero sólo siete de ellos poseen asignación de competencias completas y estos son: Quito, Cuenca, Guayaquil, Ibarra, Ambato, Manta y Loja.

Por lo tanto, se determinó que la extracción de la información se realizaría de los sitios oficiales de la agencia de tránsito de las tres principales ciudades del país como lo son: Quito “AMT”, Guayaquil “ATM” y Cuenca “EMOV” específicamente de la sección de Servicios.

#### 1.1.1 Agencia Nacional de Tránsito (ANT).

La Agencia Nacional de Tránsito (ANT) es el ente que ejecuta la función de garantizar la segura y libre movilidad terrestre por todo el país, se fundamenta en la Ley orgánica de transporte terrestre, seguridad vial y tránsito, y su principal función es la de regular todas las actividades relacionadas con el tránsito y la seguridad vial, lo cual, incluye la tramitación y control de matrículas vehiculares. (ANT, 2020)

#### 1.1.2 Agencia Metropolitana de Tránsito (AMT).

De acuerdo con lo publicado en la página oficial de la (AMT, 2020), esta institución nació mediante la Resolución Administrativa A0006 en el año 2013, y posee plena

autonomía administrativa, financiera y funcional, para ejecutar acciones que permitan controlar el transporte terrestre particular y comercial.

### 1.1.3 Autoridad de Tránsito Municipal (ATM).

Según lo publicado en la página oficial de la (ATM ®, 2020) esta institución fue creada en el 2012 por el Municipio de Guayaquil con el objetivo de establecer y ejecutar políticas que aseguren la regulación y control del tránsito terrestre para garantizar la seguridad vial y la preservación del ambiente dentro del marco legal de la nación.

En el mes de septiembre del 2020, esta agencia presentó fallos en su sitio web que lo llevaron a cerrar temporalmente sus operaciones digitales y tomar medidas para elevar su nivel de seguridad, como la instalación de un certificado digital haciendo que la página pasara de utilizar un protocolo HTTP a HTTPS como se puede observar en la Figura 1, lo que le valió a la institución el tener que emitir una disculpa pública a la ciudadanía guayaquileña (ATM ®, 2020).

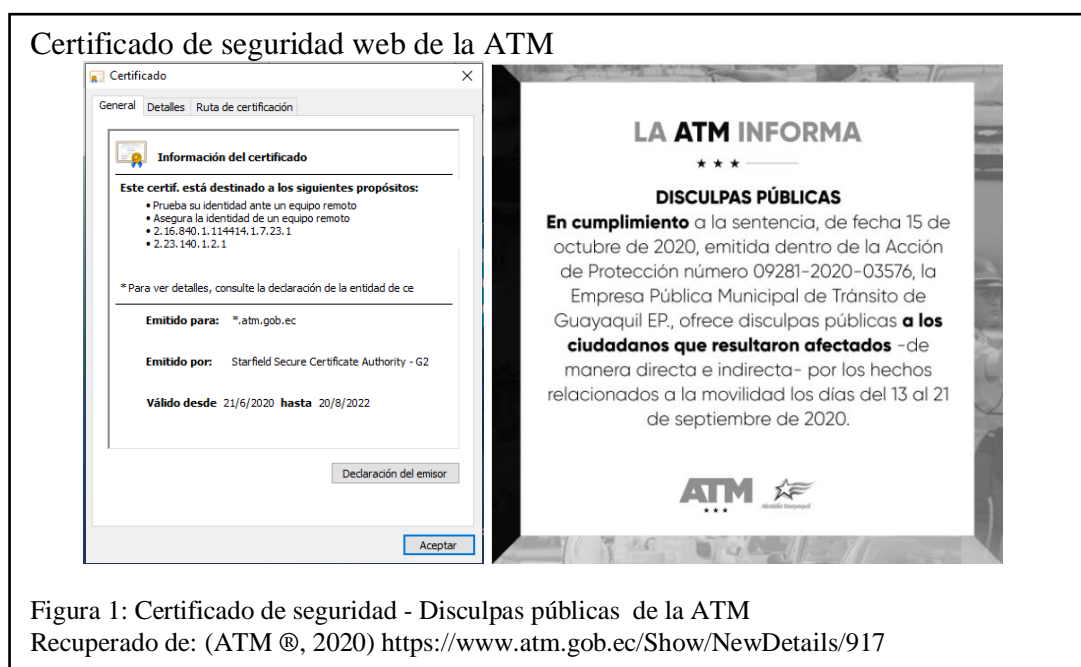


Figura 1: Certificado de seguridad - Disculpas públicas de la ATM  
Recuperado de: (ATM ®, 2020) <https://www.atm.gob.ec/Show/NewDetails/917>

#### **1.1.4 Empresa Pública Municipal de Movilidad, Tránsito y Transporte de Cuenca (EMOV).**

Esta empresa tiene como principal objetivo asegurar el correcto cumplimiento de las actividades de movilidad vehicular para aumentar el nivel de la calidad de vida, seguridad y salud pública, siempre enmarcándose en el plan de ordenamiento territorial del cantón. (EMOV, 2020).

### **1.2 Marco Teórico**

#### **1.2.1 Servidor Web.**

Según Ramos y Ramos (2011) es un conjunto de programas diseñados para recibir peticiones HTTP y con base a esta proveer el servicio de la página o conjunto de páginas web que tiene alojada. El servidor web, está en todo momento esperando por las peticiones que se efectúen de algún navegador. La respuesta emitida corresponde al envío de códigos HTML a partir de una transferencia de datos en red.

#### **1.2.2 Sitio Web.**

Según Pérez y Merino (2013) corresponde a una agrupación de páginas web, a las cuales se puede tener acceso desde un mismo dominio o subdominio de la web. Esto incluye videos, animaciones flash, documentos HTML, imágenes en diversos formatos, entre otros que puedan ser compartidos en línea.

#### **1.2.3 Certificado Digital.**

Según Giménez (2020) “Un certificado digital (X.509, v3) es un fichero que contiene (entre otros datos) la clave pública (y privada) de una persona y está avalado (firmado electrónicamente) por una entidad de confianza o Autoridad de Certificación (AC)” (p. 32).

Al tratar de acceder a un sitio web que utiliza una conexión segura, el servidor usa una identificación electrónica que está formada por una clave pública y una privada para

probar la autenticidad del sitio web a los navegadores. Estas credenciales o certificados pueden ser de diferentes tipos, tales como:

- Certificados de autoridades certificadoras: hace uso de las credenciales de una autoridad certificadora reconocida por los navegadores web.
- Certificados digitales de usuario: son utilizados para la identificación de personas naturales.
- Certificados digitales para servidores (SSL): se utilizan para probar la autenticidad de páginas web empresariales, aseguran la identidad de un servidor mediante la distribución de su llave pública.
- Certificados de software: su objetivo es brindar un software para obtener código de la web de forma segura y reducir el impacto de programas maliciosos como los virus. (Giménez, 2020, p. 42).

#### **1.2.4 Web Scraping.**

El Web Scraping según Ibáñez (2014) es un proceso vinculado a indexación web, esto se logra a través de un bot o robot; es una técnica de amplia adopción por parte de los buscadores. Se centra en transformar contenidos web de tipo no estructurados, de manera general en HTML y convertirlos a tipo estructurados los cuales se pueden almacenar y analizar a través de hojas de cálculo o un repositorio de datos local.

En la Tabla 1, se describen algunas de las ventajas según Octopus Data Inc (2020) y según Martí (2016) las desventajas del Web Scraping.

Tabla 1: Ventajas y desventajas del Web Scraping

<b>Ventajas del Web Scraping</b>	<b>Desventajas del Web Scraping</b>
La obtención de datos se realiza de forma automática: también conocido como raspado	También se puede utilizar para fines ilegales, existen Bots maliciosos



web, puede extraer datos de la web sin la intervención de una persona.	programados para examinar bases de datos y extraer información sensible de sitios web
Rapidez: disminución del tiempo y esfuerzo invertidos en la obtención de los datos.	Podría irrespetar algunos términos de uso de los sitios web.
Precisión: debido a que se elimina el factor humano en este proceso, el nivel de precisión de los datos obtenidos aumenta.	El administrador de un sitio web tiene la posibilidad de implementar varias técnicas para detener los procesos de minería de datos, tales como: El monitoreo de la cantidad de solicitudes realizadas por una determinada IP y el bloqueo de la misma, contratar servicios comerciales antibots, instalar certificados de seguridad SSL, utilizar un captcha como sistema de verificación manual, añadir entradas al fichero robots.txt.
Datos limpios y estructurados: la conversión de los datos no estructurados y semiestructurados, reorganizando la información en formatos presentables.	

Nota: en esta tabla se describen las ventajas y desventajas del Web Scraping según (Martí, 2016).

### **1.2.5 Bases de datos no relacionales.**

Según Gavilánez y Salas (2016) una base de datos no relacional (BDNR) es aquella que no sigue un esquema, es decir que los objetos de una colección trabajan bajo un criterio parecido a una tabla de una base de datos relacional, y esta tabla puede tener distintos tipos de datos. Las BDNR se enfocan en la disponibilidad más que en la consistencia de los datos, por lo cual el sistema no será consistente en cada instante del tiempo disponible; entendiendo por consistencia a la exigencia que cualquier transacción de bases de datos sólo debe cambiar datos afectados de formas permitidas.

### **1.2.6 MongoDB.**

Según Gavilánez y Salas (2016) MongoDB es una base de datos no relacional que permite almacenar los datos en documentos que son manejados a través de colecciones. Estos documentos son de tipo JSON por esta razón manejan un formato liviano y fácil de interpretar tanto para máquinas como para humanos.

### **1.2.7 Apache.**

Según Márquez, Sampedro y Vargas (2002) “es un servidor web gratuito desarrollado por el Proyecto Servidor Apache, cuyo objetivo es la creación de un servidor web eficiente y fácilmente extensible con código fuente abierto del tipo BSD” (p. 2). Posee una arquitectura modular, la cual permite que el usuario incorpore funciones nuevas a su entorno, funciona perfectamente en las versiones resientes de sistemas Linux, Unix, Windows, entre otros.

### **1.2.8 ChatBot.**

Los ChatBot son Bots generados con el fin de brindar respuesta, las cuales han sido previamente concebidas en torno a una temática en particular, básicamente es un software que envía mensajes a un interlocutor (humano), dicho mensaje es una respuesta a alguna consulta efectuada a la máquina (Charlán, 2018)

### **1.2.9 JSON.**

Según Jaramillo y Sainz (2012) “es un formato ligero de intercambio de datos, independiente del lenguaje de programación” (pág. 3). Se configura como un texto plano, de lectura, escritura y generación simple, que por lo general es más ligero que el formato XML, sus principales características son: Su independencia de un lenguaje específico, su sencillez para parsear (transformación de una entrada de texto en una estructura de datos), su carencia de definición de funciones y estructuras invisibles, no poseen validador ni son extensibles. (Jaramillo & Sainz, 2012)

### **1.2.10 OctoPerf.**

Según OctoPerf (2020) ofrece un servicio de prueba de carga de A - Z para aplicaciones web y móviles. Simula el comportamiento realista de los usuarios que navegan por una aplicación. Inicia cientos de miles de usuarios simultáneos procedentes de todo el mundo o de la propia infraestructura de servidor. Implementa métricas de monitoreo cruzado con valores de rendimiento para detectar rápidamente cuellos de botella.

### **1.2.11 OWASP Zap.**

Es un software de código abierto que permite realizar pruebas de seguridad, pertenece a la familia OWASP entidad sin fines de lucro cuyo objetivo es hacer a las aplicaciones cada vez más seguras, actualmente cuenta con una gran popularidad por ser una herramienta gratuita y por estar soportada de forma activa por un grupo de voluntarios. (OWASP, 2019)

### **1.2.12 Lenguaje de programación Python.**

Según (Molina, Loja, Zea, & Loaiza, 2016) Python es un lenguaje de programación orientado a objetos, con el que se puede codificar una amplia variedad de software, incluyendo páginas web. Se trata de un lenguaje interpretado que no necesita compilarse para poder ejecutar el código, esto aporta mayor rapidez en el proceso de desarrollo.

#### ***1.2.12.1 Características Python.***

*Propósito general:* Según (Molina, Loja, Zea, & Loaiza, 2016) todo tipo de programas pueden ser creados ya que no es un lenguaje concebido de manera específica para la web, sin embargo, a través de este se pueden desarrollar páginas.

*Multiplataforma:* Python posee varias versiones que están disponibles en muchos sistemas operativos diferentes. Cuando fue creado era para Unix, luego se volvió

compatible con el resto, con la única condición de que posea un intérprete dedicado a él.

*Interpretado:* en Python el código no debe compilarse para ser ejecutado. Realmente existe una compilación interna que se ejecuta de forma transparente para el programador. (Molina, Loja, Zea, & Loaiza, 2016)

*Orientado a Objetos:* según (Molina, Loja, Zea, & Loaiza, 2016) Python le da soporte o sustento a la programación orientada a objetos, además de ofrecer alternativas simples para la generación de programas cuyos componentes puedan ser reutilizados.

*Funciones y librerías:* este lenguaje pone a la disposición del programador muchas funciones incorporadas en sí mismo, para facilitar el tratamiento de strings, números, archivos, etc.

*Sintaxis clara:* debido a una notación indentada (con márgenes), su sintaxis no suele ser atractiva visualmente.

### ***1.2.12.2 Ventajas y Desventajas.***

Según Molina, Loja, Zea, & Loaiza (2016) las principales ventajas de este lenguaje son las siguientes:

- Estilo flexible
- Ordenado y limpio
- Cuenta con una comunidad activa
- Open Source
- Simplificado y rápido

Aunque cuenta con numerosas ventajas, es importante describir también sus desventajas:

- Problemas con hosting
- Librerías incorporadas

- Curva de aprendizaje
- No tiene identificadores protegidos.

### **1.2.13 Arquitectura MVC.**

Según Molina, Loja, Zea, & Loaiza (2016) “es un patrón de arquitectura de software”, que se fundamenta en separar los datos y principalmente lo que es la lógica de negocio del software de su representación y el módulo encargado de gestionar los eventos y las comunicaciones (p. 7).

#### ***1.2.13.1 Modelos.***

Está representada por capas en las que son trabajados los datos, estos disponen de mecanismos para el acceso a la información y actualización de su estado. En una base de datos se dispondrá de manera habitual el conjunto de datos. De tal manera en los modelos se tienen las funciones de accesos a las tablas y se podrán ejecutar comando como insertar, cargar, seleccionar, entre otros. (Molina, Loja, Zea, & Loaiza, 2016)

#### ***1.2.13.2 Vistas.***

Según Molina, Loja, Zea, & Loaiza (2016) las vistas, disponen del código de la aplicación que produce la visualización de la interfaz de usuario. Es decir, permite la renderización de los estados de la aplicación en HTML. En las vistas se tiene solamente para mostrar la salida códigos PHP y HTML

#### ***1.2.13.3 Controladores.***

Dispone de los códigos que se requieren para dar respuesta a las acciones que son solicitadas a la aplicación, tales como la visualización de elementos, ejecutar una compra, buscar información, entre otros. Esta capa actúa como enlace entre los modelos y las vistas, es respuesta a los mecanismos que sean requeridos en la implementación de una aplicación. (Molina, Loja, Zea, & Loaiza, 2016)

#### **1.2.14 Gestión de la calidad del software.**

Comprende la necesidad de establecer criterios que permitan determinar los niveles más bajos que un software debe superar para que sea considerado de calidad. Ciertamente la mayoría de las características que definen al software no pueden ser cuantificadas de forma rápida y sencilla, esto constituye un problema ya se dificulta su medición por estar definidas de forma cualitativa, debido a esto es imprescindible utilizar métricas que permitan valorar cuantitativamente cada característica dependiendo del tipo de software. (Figuroa, 2012)

#### **1.2.15 Pruebas de Software.**

Según Pressman, Roger (2010) para probar un sistema se dispone de una estrategia clásica la cual inicia con un test a la unidad, seguido de una prueba de integración y finaliza con el test de valoración del software. La prueba es un elemento de un proceso muy amplio conocido como verificación y validación (VyV).

“La verificación se refiere al conjunto de tareas que garantizan que el software implementa correctamente una función específica. La validación es un conjunto diferente de tareas que aseguran que el software que se construye sigue los requerimientos” (Pressman, R., 2010, p. 384)

##### ***1.2.15.1 Métricas.***

Según Pressman, Roger (2010) estas deben estar caracterizadas y validadas correctamente. Pueden representarse a través de principios que caracterizarán y validarán las métricas, tales como:

- El valor de la métrica debe ubicarse en un rango significativo.
- Cuando una métrica representa una característica de software que aumenta cuando ocurren rasgos positivos o que disminuye cuando se encuentran

rasgos indeseables, el valor de la métrica debe aumentar o disminuir en la misma forma (Pressman, R., 2010, p. 529).

- Deben validarse de forma empírica en una gran variedad de contextos antes de utilizarse para tomar decisiones.

### 1.2.16 Estándar ISO 9126.

Es el estándar internacional para la evaluación de la calidad de productos de software el cual dicta que “cualquier componente de la calidad puede ser descrito en términos de una o más de seis características básicas, las cuales son: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad” (Figuroa, 2012, p. 1).

Cada una de estas características se divide en un subconjunto (Figura 2) que permite profundizar en la evaluación de la calidad del software.

Preguntas Claves para responder según la característica	
<i>Características</i>	<i>Pregunta central</i>
<i>Funcionalidad</i>	¿Las funciones y propiedades satisfacen las necesidades explícitas e implícitas; esto es, el qué . . . ?
<i>Confiabilidad</i>	¿Puede mantener el nivel de rendimiento, bajo ciertas condiciones y por cierto tiempo?
<i>Usabilidad</i>	¿El software es fácil de usar y de aprender?
<i>Eficiencia</i>	¿Es rápido y minimalista en cuanto al uso de recursos?
<i>Mantenibilidad</i>	¿Es fácil de modificar y verificar?
<i>Portabilidad</i>	¿Es fácil de transferir de un ambiente a otro?

Figura 2: Preguntas Centrales de acuerdo con la característica de calidad  
Elaborado por: (Figuroa, 2012)

#### 1.2.16.1 Características Propuestas por ISO 9126.

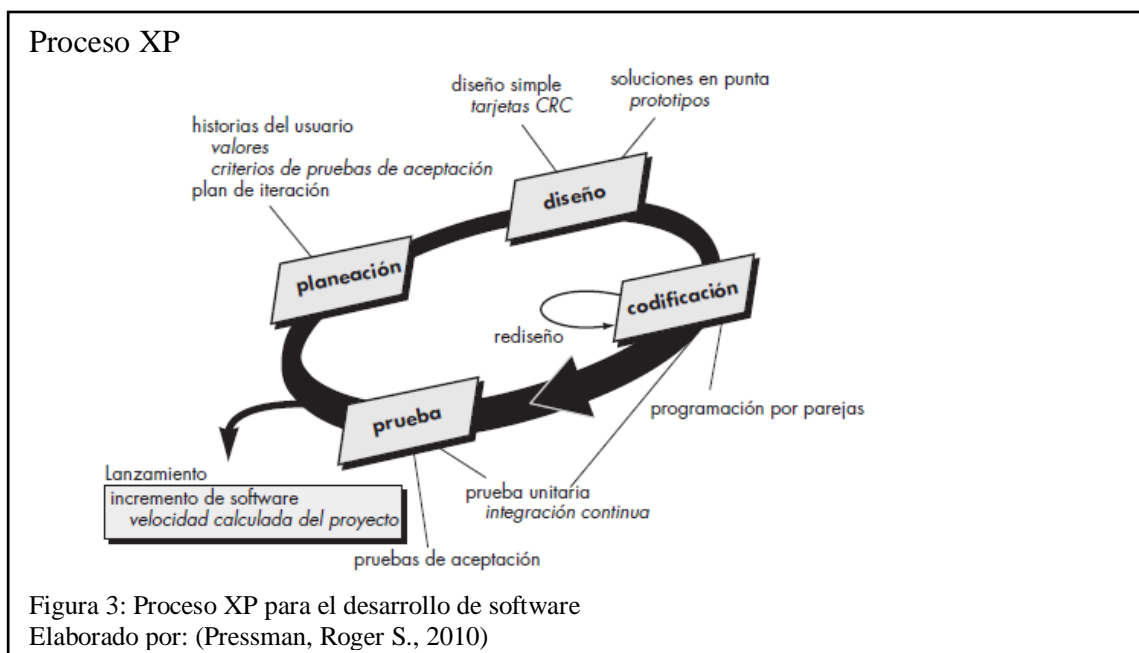
*Funcionalidad:* comprende una serie de atributos que permiten evaluar si un software ejecuta un conjunto de funciones que satisfagan las necesidades del cliente.

*Usabilidad:* permite medir el esfuerzo que el usuario final invertirá para utilizar el software.

*Eficiencia:* permite evaluar la relación entre el nivel de funcionamiento del software y la cantidad de recursos usados (Figuroa, 2012).

### 1.2.17 Metodología XP (eXtreme Programming).

Pressman (2010) Afirma que “la programación extrema usa un enfoque orientado a objetos como paradigma preferido de desarrollo, y engloba un conjunto de reglas y prácticas que ocurren en el contexto de cuatro actividades estructurales”(p. 62). El proceso de la metodología XP planteado por Pressman puede visualizarse en la (Figura 3).



#### 1.2.17.1 Características fundamentales de XP.

*Comunicación* fluida y eficaz entre el equipo de desarrollo, el cliente y otros participantes.

*Simplicidad*, la creación de un diseño sencillo y fácil de implementar en forma de código. De ser necesario, se podrá rediseñar posteriormente.

*Retroalimentación*, implementado una estrategia de pruebas eficaz se logrará por medio de sus resultados obtener una retroalimentación para el equipo de desarrollo.



*Prueba unitaria*, se realiza al finalizar la codificación de cada clase del sistema, y se ejecuta cada operación en relación a la funcionalidad establecida inicialmente (Pressman, R., 2010, p. 61).

#### **1.2.17.2 Prácticas XP.**

El proceso XP se compone por cuatro actividades estructurales: planeación, diseño, codificación y pruebas. (Pressman, R., 2010, p. 62)

*Planeación*, en esta actividad se definen los requerimientos del software, sus características y funcionalidades para que el equipo de desarrollo pueda comprender la naturaleza del negocio y las salidas o productos que deberá entregar al cliente.

*Diseño*, sigue rigurosamente el principio manténlo sencillo. XP se vale del uso de las tarjetas CRC como una herramienta eficaz para enfocarse en el software desde el contexto orientado a objetos.

*Codificación*, en esta actividad es recomendable que dos personas trabajen juntas en la creación del código fuente.

*Pruebas*, las pruebas unitarias que se crean deben implementarse con el uso de una estructura que permita automatizarlas (Pressman, R., 2010).

#### **1.2.17.3 Modelado Clase-Responsabilidad-Colaborador (CRC).**

Según Pressman, Roger (2010) “es un conjunto de tarjetas índice estándar que representan clases”; están divididas en tres secciones: en la parte superior se ubica el nombre de la clase, en la parte izquierda se enlistan las responsabilidades de la clase y en la derecha, los colaboradores (p. 148).

#### **1.2.18 Adaptación de la Metodología XP.**

Según Pressman (2010) el desarrollo ágil “reconoce que la planeación en un mundo incierto tiene sus límites y que un plan de proyecto debe ser flexible” (p. 57). En la

metodología XP es fundamental que el proceso de desarrollo se estructure de tal forma que permita al equipo desarrollador adaptar las tareas para ejecutarlas directamente, con una planeación que permita que se mantenga la fluidez necesaria, eliminando todos los procesos que no sean esenciales y enfocándose en un plan de entregas incrementales para lograr la puesta en funcionamiento del software lo más rápido posible.

Por otro lado, la metodología XP puede combinarse perfectamente con modelos de trazabilidad ágil; a pesar de que esta metodología fue diseñada para utilizar historias de usuario, el desarrollo de las aplicaciones está dirigido en su mayoría por requerimientos tradicionales que proveen precisión y facilidad de gestión al proyecto, algunas de las razones para hacer trazabilidad en proyectos ágiles son: el valor que proporciona la funcionalidad, porque los desarrolladores rara vez podrán recordar el propósito de las funcionalidades y por lo esencial que es establecer y mejorar el conocimiento del producto. (Izaurre & Andriano, 2015)

### **1.2.19 Trazabilidad Ágil.**

De acuerdo con la información obtenida de Izaurre & Andriano (2015) la trazabilidad se refiere a la capacidad de describir y seguir la vida de un requerimiento tanto en las iteraciones futuras como en las de origen, hasta su implementación.

A través de la trazabilidad, se puede determinar el origen de un requisito, su relevancia y cómo fue implementado, verificado y validado para asegurar que los mismos cuenten con su diseño, codificación y pruebas. Según Tabares, Barrera, Arroyave y Pineda (2007) los modelos de trazabilidad son aquellos que los desarrolladores crean para determinar los requerimientos, controlar su evolución y cambios en los mismos que puedan presentarse a futuro.

### **1.2.19.1 Matriz de Trazabilidad.**

La matriz de trazabilidad puede utilizarse dentro del proceso de desarrollo de software ágil con la finalidad de establecer los requerimientos y asegurarse que no existan omisiones, inconsistencias y errores previo a la construcción del software. (Pressman, R., 2010, p. 66)

Según Cuervo, Estupiñán y Álvares (2010) “esta matriz describe y sigue la vida de un requisito, permite tener una visión rápida de las relaciones de dependencia entre objetivos y requisitos, además determinar el impacto de futuros cambios en los requisitos” (p. 2).

## Capítulo 2

### Análisis y Diseño

#### 2.1. Requerimientos Funcionales y No Funcionales.

En esta sección se logró crear una matriz de trazabilidad (Anexo N° 01) para definir los requerimientos del software, los cuales se resumieron en la Tabla 2, donde también se mencionan los beneficios que dichos requerimientos aportan a ChatBot. La matriz de trazabilidad en la que se logró definir los requerimientos del sistema posee la siguiente estructura:

- *Código RF#.* y *RNF#.*: se trata de un identificador numérico y único para cada requerimiento.
- *Nombre del requerimiento*: en este campo se menciona la actividad requerida.
- *Descripción*: se realiza una breve explicación del proceso que se debe ejecutar.
- *Dependencia*: en esta sección se menciona a las actividades de las cuales depende el requerimiento en caso de que existan.
- *Riesgo y Prioridad*: como sus nombres lo indican estos campos hacen referencia al nivel de riesgo, relevancia y preferencia del requerimiento, teniendo tres posibles respuestas: alto, medio y bajo (A, M y B).
- *Objetivos*: en este renglón se mencionan los objetivos (General y específicos) que están estrechamente vinculados con el requerimiento.
- *Diseño*: describe la actividad que modela la solución correspondiente al requerimiento.
- *Desarrollo*: describe la tarea de codificación relacionada con el requerimiento.
- *Pruebas*: menciona el tipo de prueba donde se verifica y valida que se cumplió con el requerimiento.

Tabla 2: Requerimientos del sistema

<b>Código</b>	<b>Nombre del requerimiento</b>	<b>Descripción</b>	<b>Criterios de Aceptación</b>	<b>Beneficios para el Chatbot</b>
RF01	Inicio de sesión	Para ingresar al sistema se debe hacer uso de los datos de acceso: usuario y contraseña	Suministrar los datos de acceso correctamente para su validación, si no son correctos no podrá ingresar al sistema	No aplica.
RF02	Validación de URL	Se valida si se ingresó una URL de dominio principal o una URL específica del servicio.	Se valida que exista una URL de dominio principal o una URL específica del servicio.	No aplica.
RF03	Obtener información de la página web	Se deben recuperar los datos del proceso de matriculación vehicular de la página web correspondiente a la URL ingresada	Extraer automáticamente la información de servicios existentes en la URL ingresada.	Obtener automáticamente la información del proceso de matriculación vehicular.
RF04	Almacenar información en archivo JSON	Almacenar la información correspondiente al proceso de matriculación vehicular en un archivo de datos JSON	Crear el archivo JSON con toda la información extraída.	Almacenar la información de acuerdo a la estructura de la base de datos del Chatbot.
RF05	Almacenar archivo JSON en BDD	Se deberá almacenar el archivo JSON obtenido en el paso anterior en la base de datos no relacional de MongoDB	Guardar el archivo JSON creado.	La información almacenada en la base de datos puede ser extraída por el Chatbot.
RF06	Visualizar históricos	El sistema mostrará una lista ordenada de ejecuciones de Web Scraping.	Cada ejecución del Web Scraping generará un JSON. Y estos se mostrarán de forma ordenada por fecha (Ascendente)	No aplica.
RF07	Visualizar Reportes	Se mostrará en pantalla de forma más entendible los datos obtenidos en la	Cada ejecución del Web Scraping generará un reporte. Y estos se	No aplica.

		ejecución del Web Scraping	mostrarán de forma ordenada por fecha (Ascendente)	
RNF01	Facilidad de Uso	La interfaz de usuario debe ejecutarse en un ambiente web y debe ser sencilla y amigable para que se puedan manejar las diferentes opciones de una manera rápida y sin mucho esfuerzo.	Interface amigable para el usuario.	No aplica.
RNF02	Exactitud	El sistema debe extraer la información exacta para lo cual fue codificado según su algoritmo.	Verificar la información extraída vs la información de las páginas.	No aplica.

Nota: en esta tabla se describen los requerimientos del sistema y sus beneficios para el ChatBot.

### **2.1.1 Análisis de la estructura de las páginas AMT, ATM, EMOV.**

Las páginas web sobre las que se ha decidido realizar el proceso de Web Scraping son: las correspondientes a las agencias de tránsito de las ciudades Quito, Guayaquil y Cuenca.

A través del Web Scraping se logra un avance significativo en el proceso de recolección y entrega de los datos que alimentará al ChatBot, ya que se obtienen resultados de forma rápida y precisa, generando un ahorro considerable de tiempo y esfuerzo en el proceso de extracción de la información.

Antes de pasar al desarrollo del proceso de Web Scraping, se deben analizar las diferentes páginas web que contienen los datos a extraer, dicha información se encuentra en la sección de servicios de cada una de las agencias anteriormente mencionadas.

### 2.1.1.1 Agencia Metropolitana de Tránsito (AMT).

Se accede a su sitio web a través de la URL: <http://www.amt.gob.ec/> luego se ingresa en la sección SERVICIOS.



Figura 4: Sección de Servicios de la página oficial de la AMT  
Elaborado por: Página oficial AMT (<http://www.amt.gob.ec/>)

Cada una de las páginas que son fuentes de datos para el ChatBot tienen diferentes esquemas, por lo que estos se agruparon de acuerdo con la información que contienen. Para analizar la página que se muestra en la (Figura 4) se accede en primer lugar a la sección de SERVICIOS > Matriculación Vehicular > Trámites > Transferencia de dominio. Una vez seleccionado un servicio en específico (ejemplo. Transferencia de dominio), se puede observar que las páginas web contienen: el título de la página, nombre del servicio, información, importante y valores.

La información se muestra ordenada bajo estas 5 divisiones bien marcadas en la que se distribuye la información del servicio:

*Título principal:* Se encuentra en la parte superior del documento y hace referencia al título padre de toda la página. Si se analiza el código HTML, se encuentra que los títulos principales se identifican mediante etiquetas de tipo “h”.

*Título secundario:* A este se lo denominará nombre del servicio, que se encuentra inmediatamente después de un título principal y que también pueden tener una jerarquía entre ellos y para indicar que son títulos secundarios utilizan varias etiquetas propias de cada nivel de título.

*Información:* Será la parte principal del documento. Aquí se puede encontrar texto, listados o tablas que muestran los requisitos del servicio. En esta sección se puede encontrar el uso de etiquetas de tipo “li”.

*Importante:* se puede encontrar listas con información considerable del servicio, sin embargo, la información del campo importante se encuentra separada por las etiquetas de tipo “p” y “ul” las cuales permiten marcar el texto de una manera notoria.

*Valores:* aquí se puede encontrar listas con información fundamental, que indica el coste de los trámites realizados por el servicio.

A continuación, se muestra la Figura 5, donde se pueden apreciar las diferentes divisiones de la opción de servicios.

**Página Oficial de la AMT – Distribución de la información**

guro | [amt.gov.ec/index.php/servicios/matriculacion-vehicular-menu/tramites/transferencia-de-dominio.html](http://amt.gov.ec/index.php/servicios/matriculacion-vehicular-menu/tramites/transferencia-de-dominio.html)

INICIO QUIÉNES SOMOS SERVICIOS TRANSPARENCIA RENDICIÓN CUENTAS CONTRATACIONES EMERGENCIA SANITARIA CONTACTOS

**Transferencias de dominio** } **TITULO PRINCIPAL**

**TRANSFERENCIA DE DOMINIO DE VEHICULOS CON CONTRATO DE COMPRA VENTA (VENDE PERSONA CASADA)** } **TITULO SECUNDARIO**

- Original de cédula de identificación.
- Original de la matrícula. En caso de pérdida de matrícula, presentar la denuncia realizada ante la autoridad competente [www.funcojudicial.gov.ec](http://www.funcojudicial.gov.ec) la misma que debe encontrarse vigente (30 días) y el certificado único vehicular.
- Contrato de compra venta legalizado y firmado por los dos cónyuges, en caso de pérdida presentar la denuncia realizada ante la autoridad competente. [www.funcojudicial.gov.ec](http://www.funcojudicial.gov.ec), la misma que debe encontrarse vigente (30 días) y la compulsia emitida en la misma notaría que se legalizó el contrato. Si el contrato está caducado (más de 30 días), debe solicitar la citación a un agente de tránsito por contrato caducado, presentar original y copia del documento para adjuntar al trámite.
- Pago de multas asociadas a la licencia y al vehículo. (Verificar multas COIP en [www.amt.gov.ec](http://www.amt.gov.ec))
- Pago del 1% de la transferencia de dominio.
- Aprobado de revisión técnica vehicular
- Pago de matrícula del año en curso
- Levantamiento de improntas de motor y chasis (esto se lo realiza en las instalaciones de los centros de matriculación Bicentenario y Quitumbe y se levantan solo en la misma agencia donde se levantó las improntas)
  - Para el levantamiento de improntas se debe presentar el kit de seguridad (extintor, triángulos de seguridad, botiquín de primeros auxilios y llanta de emergencia).
- Pasar por revisión de valores

**INFORMACIÓN**

**IMPORTANTE:**

- Persona Natural:** El trámite puede ser realizado por un familiar directo (primer grado de consanguinidad: padres o hijos) o por el cónyuge presentando únicamente la cédula de ambas partes. En caso de ser realizado por una tercera persona se debe adjuntar el poder especial actualizado, (mínimo dos años a partir de la fecha de emisión del mismo, caso contrario, presentar la no revocatoria emitida en la misma notaría donde se obtuvo el poder)
- Persona Jurídica:** En caso de ser realizado por una tercera persona se debe adjuntar la carta de autorización suscrita por el Representante Legal de la empresa, ruc, nombramiento y copia de cédula del mismo y de la persona autorizada por el titular del vehículo.

**VALORES A CANCELAR:**

- \$22.00 dólares por concepto de DUPLICADO DE MATRICULA en el Banco del Pacífico. (AMT)
- \$7.00 dólares por concepto de TRASPASO DE DOMINIO VEHICULAR en el Banco del Pacífico. (AMT)

**VALORES**

Figura 5: Divisiones en las que se distribuye la información del servicio de la AMT  
Elaborado por: Página oficial AMT (<http://www.amt.gov.ec/>)



### 2.1.1.2 Autoridad de Tránsito Municipal (ATM).

Se accede a su página oficial (Figura 6) a través de la URL: <https://www.atm.gob.ec/> luego se ingresa en la sección SERVICIOS > TRAMITES.



Figura 6: Sección de Servicios de la página oficial de la AMT  
Elaborado por: Página oficial AMT (<https://www.atm.gob.ec/>)

En la (Figura 7) se presenta la página web que contiene el título de la página que se utilizará como nombre del servicio, información, y consideraciones. La información se muestra ordenada bajo estas 3 divisiones bien marcadas en la que se distribuye la información del servicio:

*Título principal:* Se encuentra en la parte superior del documento y hace referencia al título de toda la página, estos se identifican mediante etiquetas de tipo “h”.

*Información:* Será la parte principal del documento. Aquí se puede encontrar texto, listados o tablas que muestran los requisitos del servicio.

*Consideraciones:* al igual que en la sección anterior aquí se puede encontrar listas o párrafos con información considerable del servicio.

## Página Oficial de la ATM – Distribución de la información



Figura 7: Divisiones en las que se distribuye la información del servicio de la ATM  
Elaborado por: Página oficial ATM (<https://www.atm.gob.ec/>)

### 2.1.1.3 Empresa Pública Municipal de Movilidad, Tránsito y Transporte de Cuenca (EMOV).

Todos los servicios que ofrece esta entidad direccionan hacia un formulario, archivo PDF, como se observa en la (Figura 8) y se accede a ella a través de la URL: <https://www.emov.gob.ec/> luego se ingresa en la sección SERVICIOS.

## Página Oficial de la EMOV – Sección servicios

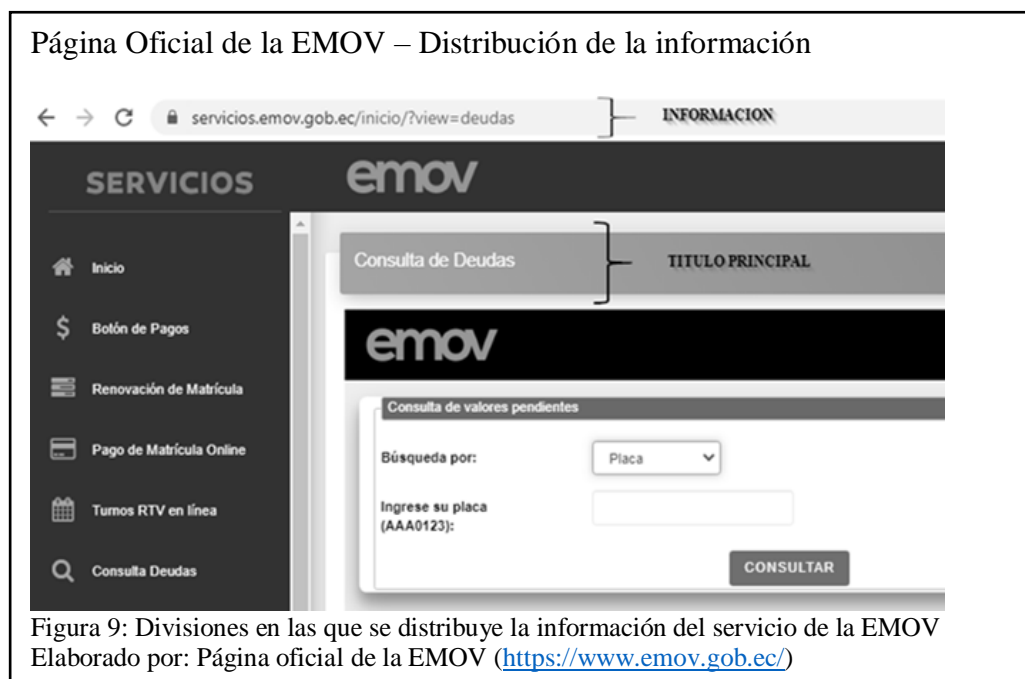


Figura 8: Sección de Servicios de la página oficial de la EMOV  
Elaborado por: Página oficial de la EMOV (<https://www.emov.gob.ec/>)

La información se muestra ordenada bajo 2 divisiones bien marcadas en la que se distribuye la información del servicio como se observa en la (Figura 9).

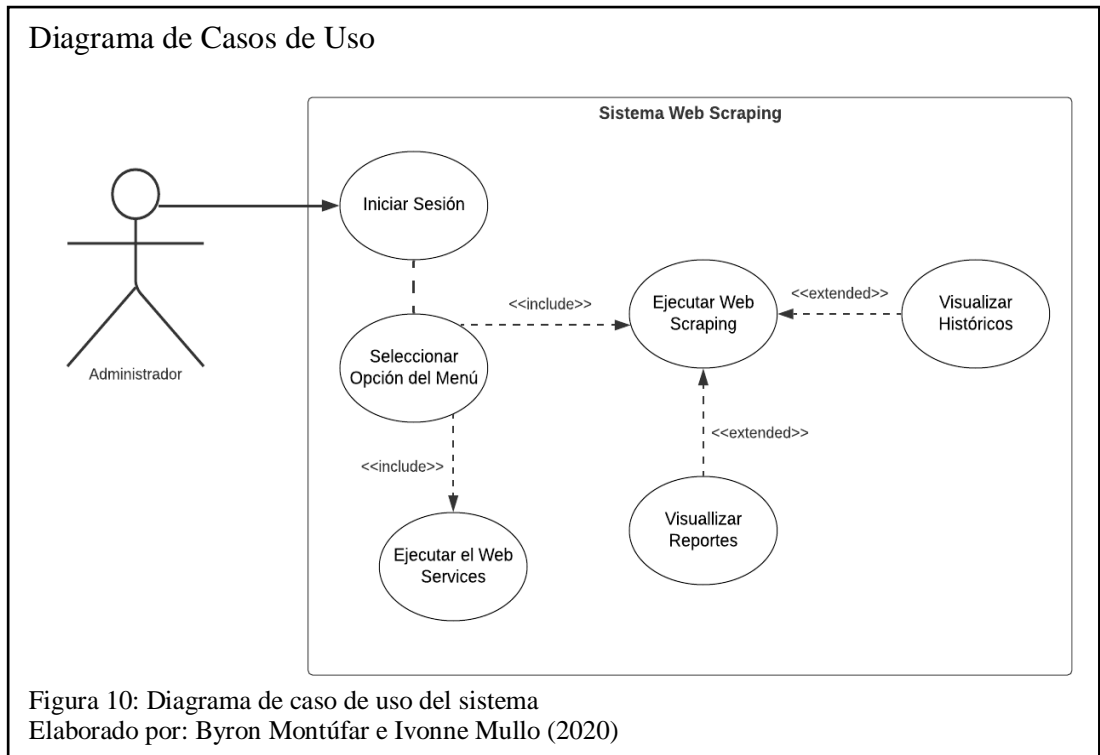
*Título principal:* Se encuentra en la parte superior del documento y hace referencia al título padre de toda la página.

*Información:* Se mostrará la URL de los formularios, páginas web o archivos PDF estos se encuentran dentro de las etiquetas de tipo “href” las mismas dentro de una lista.



### 2.1.2 Diagrama de Caso de Uso de la aplicación Web.

Para desarrollar el diagrama de la (Figura 10), se analizó la manera a través de la cual se deberían realizar los procesos, se identificó el actor involucrado posteriormente se logró determinar un caso de uso generalizado, donde se muestran las acciones que pueden ser realizadas por el actor Administrador.



### 2.1.3 Actor.

En esta sección se describe el actor principal del sistema, tomando en cuenta que un actor es cualquier cosa (usuario, máquina, programa, etc.) que se comunique directamente con el sistema y que sea externo a éste (Pressman, R., 2010) .

Tabla 3: Actor del sistema

Actor	Descripción
 Administrador	AN01: Administrador Está representado por la persona que podrá realizar las siguientes acciones en el sistema: Ejecución del Web Scraping (extracción de la información de las páginas de la AMT, ATM y EMOV) Consulta de históricos. Visualización de reportes

Nota: En esta tabla se describen las funciones del Actor del sistema (Administrador).

Todo actor tiene un motivo cuando ingresa a un sistema, en la Tabla 3, se describieron las funcionalidades que pueden ser ejecutadas por el actor Administrador.

#### **2.1.4 Diagrama de actividades del sistema.**

En la (Figura 11) se presentan todas las actividades que pueden ser realizadas por el usuario (Administrador), desde el inicio de sesión a través de la validación de usuario y contraseña, la selección de un módulo para ser ejecutado (Web Scraping, Histórico ó Reportes), una vez que el usuario selecciona el módulo que le interesa ejecutar consigue una serie de pasos a realizar de acuerdo al resultado que desea obtener del software. Este diagrama fue diseñado usando lenguaje UML y se realizó lo más sencillo posible para lograr una mayor comprensión del software.

### Diagrama de actividades del sistema

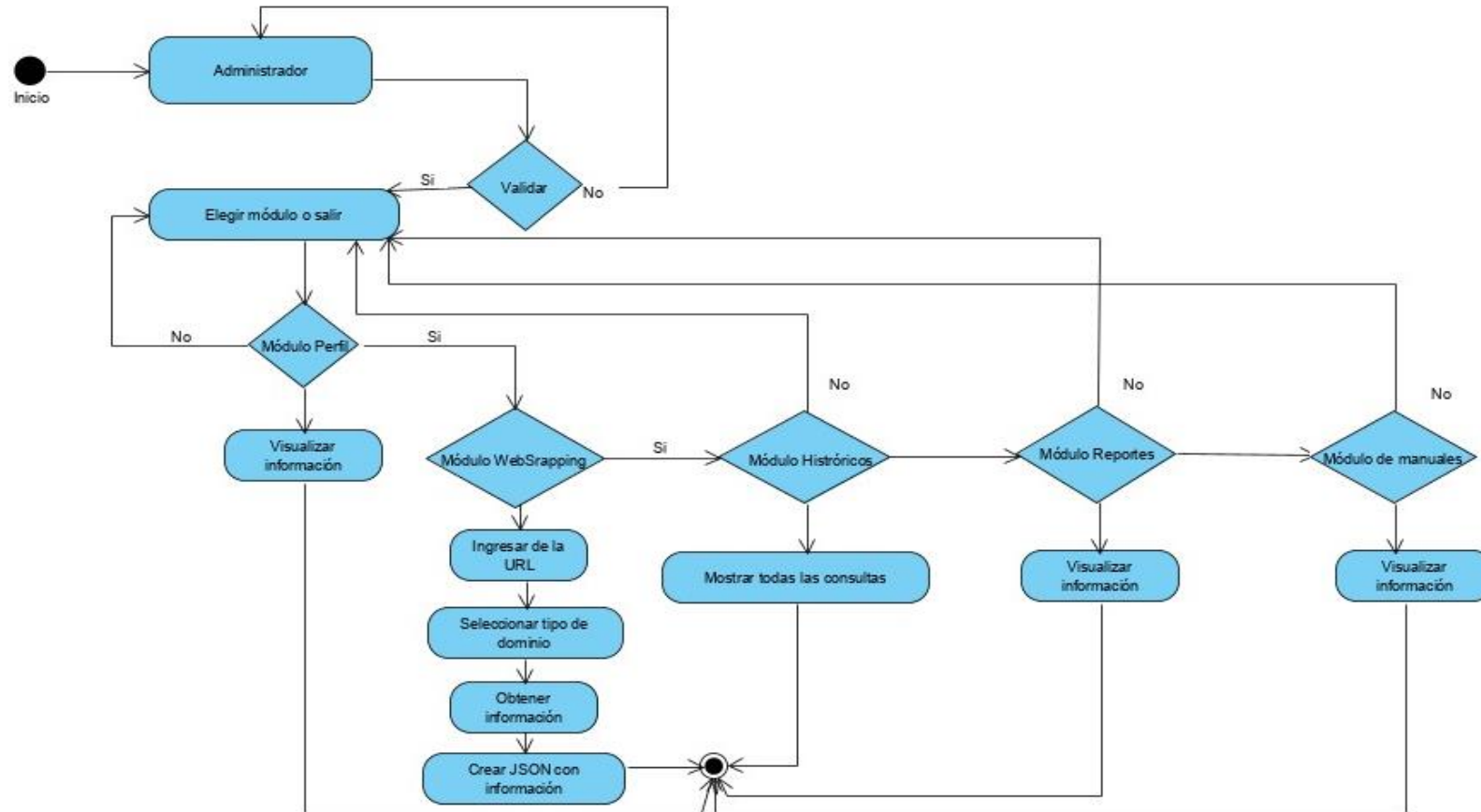
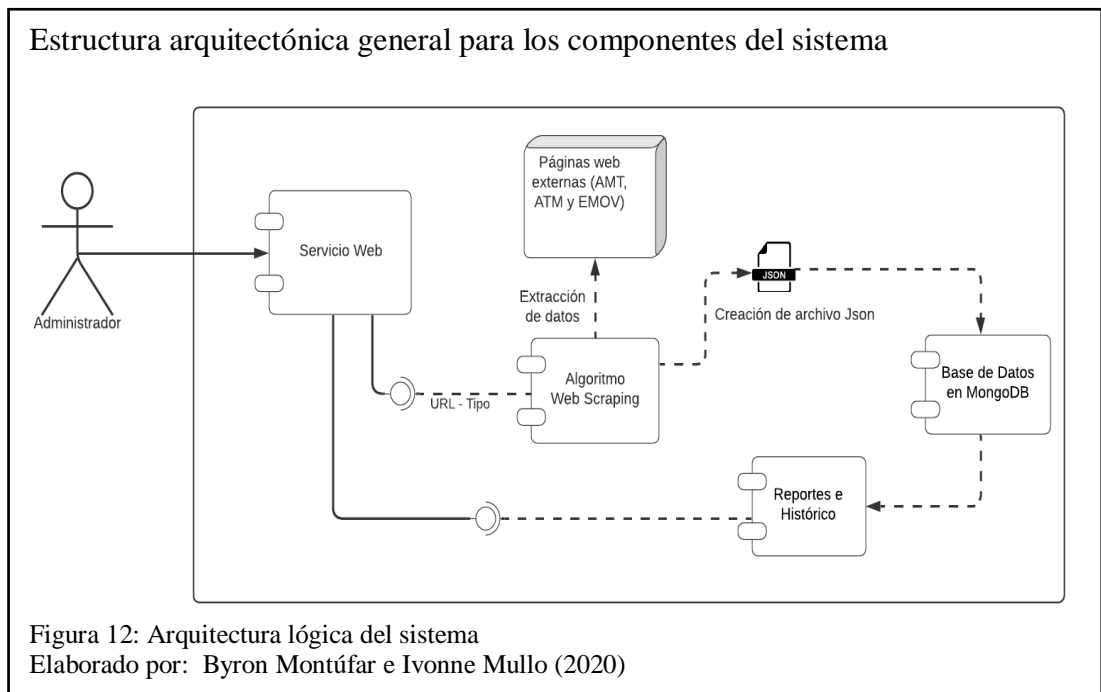


Figura 11: Diagrama de actividades del sistema  
Elaborado por: Byron Montúfar e Ivonne Mullo (2020)

### 2.1.5 Arquitectura lógica del sistema.

El sistema se encuentra estructurado en base al modelo MVC y está comprendido por los siguientes componentes: el servicio web, la consulta para la autenticación de usuario, el algoritmo de Web Scraping, todos ellos codificados en Python, adicionalmente una base de datos en MongoDB para el almacenamiento de los archivos de tipo JSON y dichos archivos que se encargan de almacenar y transferir la información. Tanto el lenguaje de programación utilizado como el manejador de base de datos son de código abierto y poseen licencia de libre distribución.



En la (Figura 12) se muestra la relación estructural entre los componentes del sistema, utilizando los elementos del modelo de componentes UML y los íconos de las herramientas de software utilizadas en cada uno. Comenzando por el componente del menú principal desde donde se ejecutan todas las funciones del sistema, a través de la interfaz de usuario se envía la URL objetivo y el tipo de dirección (de sitio principal o de servicio) al componente de ejecución del algoritmo de Web Scraping, el cual, ejecuta la extracción de datos de la página web externa, crea el archivo JSON con los

datos recopilados y almacena dicho archivo en la Base de datos. Accediendo a través de la interfaz de usuario de Visualizar Reportes y Visualizar Histórico se puede acceder a los archivos JSON generados previamente

### 2.1.6 Modelado Clase-Responsabilidad-Colaborador (CRC).

En las siguientes cinco tarjetas CRC se muestran las clases utilizadas dentro de la codificación del algoritmo de Web Scraping, la acción o función que ejecuta cada una de ellas y las clases colaboradoras o de las que dependen.

Tarjeta CRC# 01:

<b>Clase:</b> Query	
<b>Responsabilidad:</b>	<b>Colaborador:</b>
Guarda Inicio de Sesión	Conexion2

Tarjeta CRC# 02:

<b>Clase:</b> Scrapyweb	
<b>Responsabilidad:</b>	<b>Colaborador:</b>
Extrae Datos de la Página web	Servicescrap - Conexion

Tarjeta CRC# 03:

<b>Clase:</b> Servicescrap	
<b>Responsabilidad:</b>	<b>Colaborador:</b>
Guarda Archivo JSON	Conexion

Tarjeta CRC# 04:

<b>Clase:</b> Conexión	
<b>Responsabilidad:</b>	<b>Colaborador:</b>
Conectar a la base de datos bdd_webscrapping. Colección: bdd_webscrapping	Servicescrap

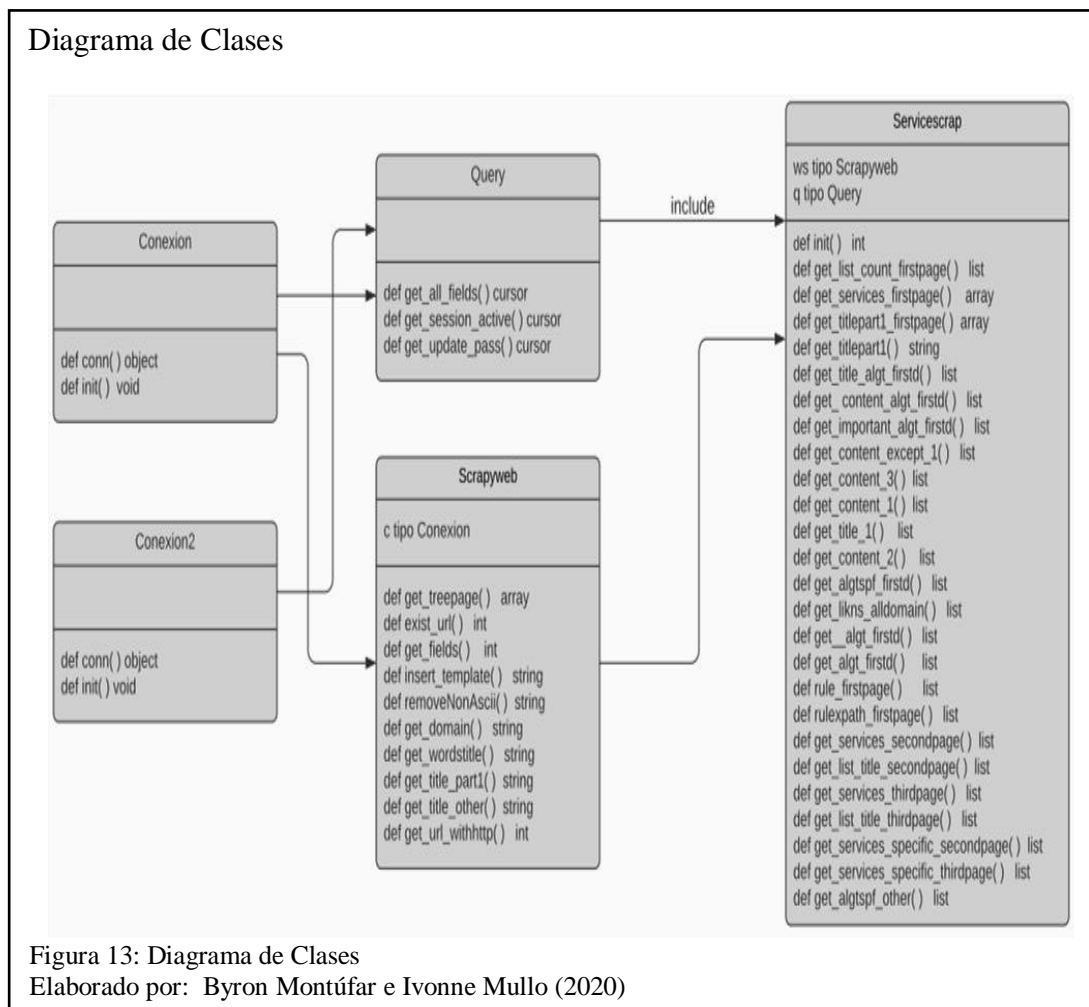


Tarjeta CRC# 05:

<b>Clase: Conexión2</b>	
<b>Responsabilidad:</b>	<b>Colaborador:</b>
Conectar a la base de datos bdd_webscrapping. Colección: bdd_user	Query

### 2.1.7 Diagrama de clases.

El siguiente diagrama en lenguaje UML (Figura 13) describe las clases que se crearon en el proceso de codificación del software con sus respectivos campos de datos, atributos, operaciones y las relaciones con las demás clases colaboradoras.



### **2.1.8 Estructura de la base de datos no relacional.**

En la consola de MongoDB y tomando como referencia las necesidades expresadas por el equipo de desarrollo del ChatBot, se creó la base de datos no relacional llamada bdd\_wescrapping, posteriormente se creó la colección bdd\_user que se utiliza para el inicio de sesión, y la colección bdd\_webscrapping para almacenar los archivos JSON resultantes del proceso de extracción de datos del Web Scraping y así lograr que se encuentren disponibles para futuras consultas, esta base de datos por su naturaleza no relacional carece de una estructura como tablas y campos.

#### ***2.1.8.1 Esquema conceptual del archivo JSON.***

La estructura del archivo JSON fue diseñada de acuerdo a los requisitos de datos de entrada para el ChatBot y contiene los parámetros que se detallan a continuación:

- "servicio": en este campo se almacena el título del servicio
- "importante": en este campo se almacena toda la información relevante que tiene cada servicio, cabe mencionar que no todos los servicios tienen esta información relevante, en ese caso esta sección se deja en blanco.
- "URL": muestra la URL de donde obtuvo la información
- "identificador": este se compone de la extracción de las primeras letras de cada palabra empezando desde el "menú" en donde se encuentra, más el título del servicio (solo se toman en cuenta las palabras que tienen de 3 letras en adelante).
- "información": contiene los requisitos que se detallan en cada servicio, necesarios para realizar el trámite en cuestión.
- "valores": es la información de los montos a cancelar (pago) por el usuario de acuerdo al servicio que desea solicitar.

Una vez que el archivo JSON ha sido construido, este se almacena en orden descendente de acuerdo a la fecha de su generación, en una BDNR creada en

MongoDB, con la finalidad de que se encuentre disponible para las consultas que se realizan a través de las funcionalidades de Reportes e Históricos del software.

## Capítulo 3

### Construcción y Pruebas

#### 3.1 Estándares de programación de Python

Aun cuando el código de programación de un software esté escrito y estructurado de forma correcta y ordenada, es necesario usar estándares, estilos o convenciones que ayuden al desarrollador a crearse buenos hábitos de programación para comprender y documentar su software. (Varsovia, Hylton, Goodger, & Coghlan, 2020)

Dicho esto, se establecen varias convenciones como (denominaciones, formatos, etc.) para la escritura de código. Los cuales fueron aplicados mediante las buenas prácticas de programación en Python. Entre ellas, se encuentran:

- PEP 20 basado en “legibilidad cuenta”
- PEP 3131 Codificaciones, Importaciones
- PEP 257 Cadenas de Documentación

#### Estándar PEP20 – Principios de organización del código

Para ordenar código es importante aplicar espacios por indentación, como se observa en el ejemplo de la (Figura 14), las líneas de codificación deben ser alineadas verticalmente con el carácter que se haya utilizado.

Fragmento de código

```
if sessionactive.count() > 0:
    for x in sessionactive:
        usuario = x['Usuario']
        rolusu = x['Rol']
    return render_template('profile.html', nomuser = usuario, rol = rolusu)
```

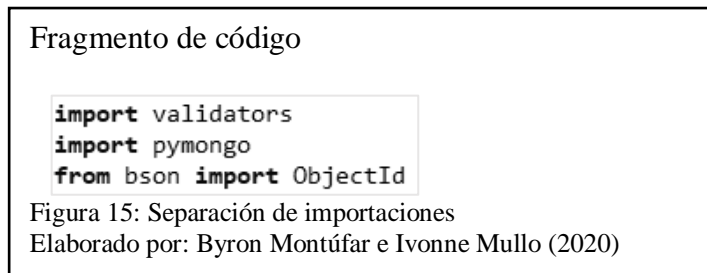
Figura 14: Ejemplo de indentación

Elaborado por: Byron Montúfar e Ivonne Mullo (2020)

Dichas líneas de código pertenecen al inicio de sesión de usuarios que realizan para el respectivo ingreso a la aplicación web.

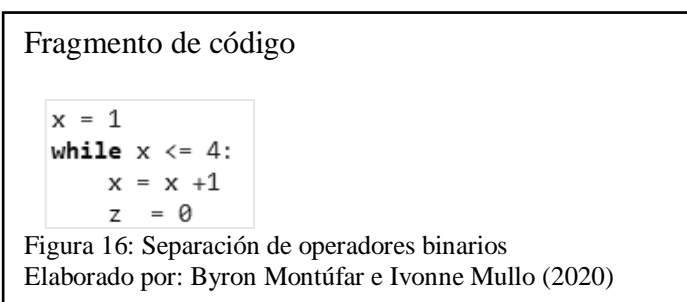
## Estándar PEP 3131 – Principios para codificaciones e Importaciones

Las librerías prescritas están codificadas con caracteres correspondientes a la codificación ASCII, y se usa palabras en inglés. Las importaciones se encuentran debidamente separadas como se visualiza en el ejemplo de la (Figura 15), líneas de código perteneciente a query.py.



## Separaciones en Operadores Binarios

Al momento de codificar Operadores Binarios se aplica el respectivo espacio en cada lado, manteniendo la misma cantidad de espacios como se muestra en la (Figura 16) donde se pueden ver las líneas de código pertenecientes a servicescrap.py, operadores binarios para dar inicio a la aplicabilidad del algoritmo.



## Comentarios

Forman parte fundamental dentro de la codificación, porque tiene como finalidad explicar brevemente utilizando ternarios y diferentes funciones parte del código, como se observa en la (Figura 17). Es recomendable aplicarlos porque servirá como base para facilitar el proceso de desarrollo a posteriores compañeros que continúen con el mejoramiento de la aplicación web.

### Fragmento de código

```
data['pagecontent'] = []
#links para servicios http://www.amt.gob.ec/
#selecciono el xpath de servicios
elementos = xml_tree.xpath('/html/body/div[1]/div/section[1]/div/div/div/ul/li[3]/descendant::*[@href]')
#declaro una variable para almacenar las urls de los servicios
list_services = []
#recorro el arbol de elementos
for li in elementos:
    #verifico que todos los href tengan contenido
    if li.get('href') != '#':
        #formo el url con un formato correcto
        if li.get('href').startswith('http') or li.get('href').startswith('https'):
            link = li.get('href')
        else:
            link = "http://www.amt.gob.ec"+li.get('href')
        #almaceno cada elemento a la lista
        list_services.append(link)
```

Figura 17: Ejemplo de comentarios

Elaborado por: Byron Montúfar e Ivonne Mullo (2020)

## Estándar PEP 257 - Principios de nombramiento

Las tildes no se encuentran incluidas. Cabe mencionar que en toda la codificación se implementa nombramientos tanto para llamado de clases, funciones, sentencias, etc.

Se aplica todo en minúscula, adicional, con letras en minúscula con guiones bajos.

### Fragmento de código

```
for x in range(len(list_inver)):
    self.si_cubre_2 = self.si_cubre_2 + 1
    list_title = list_inver[::-1]
else:
    list_title = self.get_content_except_1(tree,desc_list)
```

Figura 188: Ejemplo de nombramiento

Elaborado por: Byron Montúfar e Ivonne Mullo (2020)

En la (Figura 18) se observan las líneas de código pertenecientes a `servicescrap.py`, usadas para obtener rango de números de la lista, donde se selecciona el primer y último elemento de la lista depurada.

## Nombres de clases

Para los nombres de las clases se utilizó la convención “Cap Words” (palabras que comiencen con mayúsculas).

### Fragmento de código

```
class Servicescrap():
```

Figura 19: Ejemplo de nombres de las clases

Elaborado por: Byron Montúfar e Ivonne Mullo (2020)

En la (Figura 19) se observan las líneas de código pertenecientes a webscrap.py, permite extraer la URL de cualquier página.

### Argumentos de funciones y métodos

El uso de self para el primer argumento de los métodos de instancia, como se muestra en el ejemplo de la (Figura 20).

Fragmento de código

```
def exist_url(self, url):
    std = 1
    validators.url(url)
    if not validators.url(url):
        std = 0
    return std
```

Figura 20: Líneas de código pertenecientes a webscrap.py.  
Elaborado por: Byron Montúfar e Ivonne Mullo (2020)

### 3.2 Construcción de la base de datos no relacional

En cuanto a la construcción de la base de datos tipo no relacional, inicialmente se ejecutan comandos como root, en el ejemplo de la (Figura 21) en ese momento se inactivo. Para verificar el estado del servicio de la base de datos (MongoDB) se ejecuta la sentencia:

- systemctl status mongod

Estado del servicio de base de datos

```
root@webscrap:~# systemctl status mongod
● mongod.service - MongoDB Database Server
   Loaded: loaded (/lib/systemd/system/mongod.service; disabled; vendor preset:
   Active: inactive (dead)
     Docs: https://docs.mongodb.org/manual

jul 20 19:43:46 webscrap systemd[1]: /lib/systemd/system/mongod.service:10: PIDF
```

Figura 21: Estado del servicio de la base de datos  
Elaborado por: Byron Montúfar e Ivonne Mullo (2020)

Posteriormente, se procede a levantar el servicio de MongoDB (Figura 22).

## Levantamiento el servicio de MongoDB

```
root@webscrap:~# service mongod start
root@webscrap:~# █
```

Figura 22: Levantamiento el servicio de MongoDB  
Elaborado por: Byron Montúfar e Ivonne Mullo (2020)

Se verifica que el servicio se encuentre levantado correctamente volviendo a ejecutar el comando que se encuentra en la (Figura 23).

## Estado del servicio de la Base de datos

```
● mongod.service - MongoDB Database Server
   Loaded: loaded (/lib/systemd/system/mongod.service; disabled; vendor preset: enabled)
   Active: active (running) since Mon 2020-07-20 19:48:30 -05; 3min 1s ago
     Docs: https://docs.mongodb.org/manual
   Main PID: 2739 (mongod)
    Memory: 202.7M
    CGroup: /system.slice/mongod.service
           └─2739 /usr/bin/mongod --config /etc/mongod.conf

jul 20 19:48:30 webscrap systemd[1]: Started MongoDB Database Server.
```

Figura 23: Estado del servicio de la base de datos  
Elaborado por: Byron Montúfar e Ivonne Mullo (2020)

En cuanto a la conexión realizada con la base de datos, como se puede apreciar en la (Figura 24) la clase creada es (conexion.py), seguidamente se procede a importar una librería para utilizar los complementos MongoDB en Python, como:

## Fragmento de código

```
import pymongo

class Conexion:
    mycol = None
    def __init__(self):
        #apunta al servidor de la base de datos (localhost)
        myclient = pymongo.MongoClient("mongodb://localhost:27017/")
        #nombre de la base de datos
        mydb = myclient["bdd_webscrapping"]
        #nombre de la colleccion
        self.mycol = mydb["bdd_webscrapping"]

    def conn(self):
        return self.mycol
```

Figura 24: Importación de librería componentes MongoDB  
Elaborado por: Byron Montúfar e Ivonne Mullo (2020)



Luego, para conectar a la base de datos se ejecuta el servicio de MongoDB, como se muestra en la (Figura 25) con el comando mongo, dicho comando permite iniciar al cliente dentro del mismo sistema en el que se ejecuta la aplicación MongoDB.

Fragmento de código

```
debian@vps-ccc3e8e0:~$ mongo
```

Figura 25: Ejecución del comando mongo  
Elaborado por: Byron Montúfar e Ivonne Mullo (2020)

Se selecciona la base de datos creada, ejemplo en la Figura 26, con el comando use bdd\_webscrapping.

Fragmento de código

```
> use bdd_webscrapping  
switched to db bdd_webscrapping
```

Figura 26: Ejecución del comando use bdd\_webscrapping  
Elaborado por: Byron Montúfar e Ivonne Mullo (2020)

La base de datos MongoDB, trabaja con colecciones, donde se va a alojar toda la información a manejar. Por tales razones, se enlista las colecciones mediante show collections como se observa en el ejemplo de la (Figura 27). Para lo cual, en el desarrollo del proyecto se usa el mismo nombre de la base de datos.

Fragmento de código

```
> show collections;  
bdd_user  
bdd_webscrapping
```

Figura 27: Ejecución del comando show collections  
Elaborado por: Byron Montúfar e Ivonne Mullo (2020)

En la (Figura 28) se muestra un ejemplo para enlistar la información que tiene la colección donde se utiliza la siguiente sintaxis: db.bdd\_webscrapping.find().pretty();

## Fragmento de código

```
{
  "servicio" : "La Revisión Técnica Vehicular es un conjunto inspecciones de un vehículo destinadas a:",
  "importante" : "La Revisión Técnica Vehicular es un conjunto inspecciones de un vehículo destinadas a:",
  "url" : "http://www.amt.gob.ec/index.php/servicios/revision-tecnica/revision-tecnica-vehicular.html",
  "identificador" : "PPRTVRTVCIVD",
  "informacion" : [
    "La Revisión Técnica Vehicular es un conjunto inspecciones de un vehículo destinadas a:"
  ],
  "valores" : "La Revisión Técnica Vehicular es un conjunto inspecciones de un vehículo destinadas a:"
},
```

Figura 28: Proceso de enlistar información

Elaborado por: Byron Montúfar e Ivonne Mullo (2020)

### 3.3 Métodos

En la Tabla 4, se describen todos los métodos utilizados en cada clase de la aplicación web.

Tabla 4: Métodos aplicación web

<b>Métodos</b>		
<b>Métodos aplicación web</b>		
<b>Nombre Método</b>	<b>Pertenece a la clase</b>	<b>Propósito</b>
def login()	logout.py	Aplica autenticación del usuario.
def to_pretty_json()	history.py	Devuelve dato en formato JSON.
def profile()	profile.py	Permite al usuario visualizar la información con la cual se realizó la autenticación.
def _init_()	conexion.py	Para establecer conexión con la base de datos (JSON)
def _init_()	conexion2.py	Para establecer conexión con la base de datos de usuarios
def get_all_fields()	query.py	Devuelve la información de la hora, fecha, estado y URL en la cual se ejecuta la consulta
def get_session_active()	query.py	Devuelve el estado de la sesión del usuario
def get_update_pass()	query.py	Edición de contraseñas de usuarios.

def get_titlepart1()	servicescrap.py	Separación bloques en HTML/head/title
def get_services_firstpage() def get_services_secondpage() def get_services_thirdpage()	servicescrap.py	Despliegue de servicios de las páginas web (Quito, Guayaquil, Cuenca)
def get_title_algt_firstd()	servicescrap.py	Definición de reglas para obtener el título
def get_content_algt_firstd()	servicescrap.py	Definición de reglas para obtener el contenido
def get_important_algt_firstd()	servicescrap.py	Definición de reglas para obtener importante
def get_values_algt_firstd()	servicescrap.py	Definición de reglas para obtener valores
def get_content_except_1()	servicescrap.py	Definición de reglas para las excepciones de las páginas
def get_likns_alldomain()	servicescrap.py	Despliegue evaluación algoritmo.
def get_algtspf_other() def get_algt_firstd() def get_algtspf_firstd()	servicescrap.py	Despliegue Web Scraping (fecha, hora, URL, identificador, importante, información, servicio, valores).
def get_list_title_secondpage def get_list_title_thirdpage	servicescrap.py	Despliegue de lista de títulos de la primera, segunda y tercera página web.

def get_treepage	webscrap.py	Transformación de las páginas en código html
def exist_URL	webscrap.py	Validación URLS
def insert_template	webscrap.py	Indexación de plantilla (template)
def removeNonAscii	webscrap.py	Remover códigos ascii
def get_domain	webscrap.py	Extracción del dominio
def get_wordstitle	webscrap.py	Permite obtener las primeras letras del nombre del servicio
def get_URL_withhttp	webscrap.py	Permite verificar si la URL es http o https

Nota: Esta tabla contiene todos los métodos de la aplicación web.

### **3.4 Proceso de extracción de datos de las páginas AMT, ATM, EMOV**

Debido a la longitud y complejidad del algoritmo programado para la extracción automática de datos de las páginas anteriormente mencionadas, en el (Anexo N° 03 hasta el Anexo N° 07) se presentan partes del mismo, con la finalidad de mostrar gráficamente cómo se realiza el proceso de extracción de datos de las páginas web de las agencias de Quito, Guayaquil y Cuenca, ingresando la URL del dominio, para facilitar su comprensión se dividió el algoritmo en 7 pasos, los cuales corresponden a:

- Paso 1: Obtener todos los links de servicio
- Se obtiene la URL de los servicios de la página web analizada, en el (Anexo N° 03) se muestra un ejemplo del flujo del algoritmo.
- Paso 2: Obtener sección títulos de servicios
- Se obtiene el título de cada servicio de la página web analizada, en los (Anexos N° 04 y 05) se muestra un ejemplo de este proceso.
- Paso 3: Función contenido 1
- Se buscan los elementos hermanos a la regla ingresada incluyendo elementos que contengan imágenes que cumplan con la condición de que se encuentren dentro de un artículo y una sección, en el (Anexo N° 06) se muestra un ejemplo del flujo del algoritmo.
- Paso 4: Función contenido 2
- Se busca los elementos hermanos a la regla ingresada incluyendo elementos que contengan imágenes y párrafos, en el (Anexo N° 07) se muestra un ejemplo este proceso.
- Paso 5: Obtener sección contenido de servicios
- Se extrae los datos referentes al contenido de cada servicio (información).
- Paso 6: Obtener sección importante de servicios

- Se extraen los datos referentes al contenido de cada servicio (sección de información importante).
- Paso 7: Obtener sección valores de servicios
- Se extraen los datos referentes al contenido de cada servicio (sección de valores).

Cabe destacar que sólo se diagramaron los 4 primeros pasos, así mismo los pasos del 5 al 7 son muy parecidos al paso 2 obtener sección títulos de servicios.

El proceso por URL específica se diferencia del anterior por las siguientes características:

- Ingresan directamente a proceso para seleccionar título
- Omite función Obtener todos los links de servicio

A continuación, se despliegan los diagramas correspondientes al algoritmo de Web Scraping por URL de dominio en sus primeros 4 pasos o divisiones de procesos.

### **3.5 Código Relevante**

#### **3.5.1 Codificación Algoritmo de Web Scraping.**

En cuanto a la programación del algoritmo, la estructuración lógica planteada se presenta en las siguientes codificaciones, ver (Figuras 29, 30 y 31), ya que la longitud total del mismo no permite su completa incorporación.

El proceso inicia desde la toma de datos para extraer la data, de hecho, en la (Figura 29) se muestra el fragmento de código para la extracción de la información de cada sesión que compone al archivo JSON (identificador, servicio, información, importante, valores, URL), esta información extraída es almacenada en listas diferentes para cada sección.

Fragmento de código para la extracción de la información de cada sección (identificador, servicio, información, importante, valores)

```
#tomo datos de titulo del arbol de la url
list_title = self.get_title_algt_firstd(tree_url,serv)
#tomo datos de contenido del arbol de la url
list_info = self.get_content_algt_firstd(tree_url,serv)
#tomo datos de importante del arbol de la url
list_important = self.get_important_algt_firstd(tree_url,serv)
#tomo datos de valores del arbol de la url
list_values = self.get_values_algt_firstd(tree_url,serv)

#list_info = self.get_list_info(tree_url)
for title,info,important,values in zip(list_title,list_info,list_important,list_values):
    i = i+1
    title_f = title.replace('\r', '').replace('\n', '').replace('\t', '')
    identify = ws.get_wordstitle(title_f)
    title_part1 = self.get_titlepart1_firstpage(tree_url)
    informacion = info.replace('\n', '')
    informacion = informacion.split('\r')
    data['pagecontent'].append({
        'identificador': ''+title_part1.upper()+''+identify.upper(),
        'servicio': ''+ title_f,
        'informacion': informacion,
        'importante': ''+important.replace('\r', '').replace('\n', ''),
        'valores': ''+values.replace('\r', '').replace('\n', ''),
        'url': ''+ serv})
else:
    i = i+1
    data['pagecontent'].append({
        'identificador': ''+title_part1.upper()+''+identify.upper(),
        'servicio': ''+ title_f,
        'informacion': 'Url con formato incorrecto',
        'importante': ''+important.replace('\r', '').replace('\n', ''),
        'valores': ''+values.replace('\r', '').replace('\n', ''),
        'url': ''+ serv})
```

Figura 29: Codificación Algoritmo de Web Scraping - Extracción de la información  
Elaborado por: Byron Montúfar e Ivonne Mullo (2020)

Fragmento de código para guardar la información extraída

```
#genero arbol tags
tree = ws.get_treepage(url)
#lista de servicios firstpage
services = serv.get_algt_firstd(tree,url)
result = services
with open('servicios_page.json', 'w') as file:
    json.dump(result, file,sort_keys=True, indent=4)
#insertar datos
data = open("servicios_page.json")
x = json.load(data)
result_id = mycol.insert_one(x)
myclient.close()
```

Figura 30: Codificación Algoritmo de Web Scraping - Guardar la información extraída  
Elaborado por: Byron Montúfar e Ivonne Mullo (2020)



En la (Figura 30) se puede observar un fragmento de código donde se configura el proceso de salvaguardar la data extraída, mientras que en la (Figura 31) se muestra la fracción de código para leer el contenido del JSON almacenado en la colección de la base de datos.

#### Fragmento de código para leer el contenido del JSON almacenado

```
function see_json(dato0,dato1){
    var request;
    if (window.XMLHttpRequest) {
        request = new window.XMLHttpRequest();
    } else {
        request = new window.ActiveXObject("Microsoft.XMLHTTP");
    }
    request.open("GET", "/Sp_webscrn/part_code.py?dato0=" + encodeURIComponent(dato0)+"&dato1="+dato1, true);
    params = 'x=' + dato1
    request.send(params);
    request.onreadystatechange = function()
    {
        }
    }
}
```

Figura 31: Codificación Algoritmo de Web Scraping - Guardar la información extraída  
Elaborado por: Byron Montúfar e Ivonne Mullo (2020)

## 3.6 Pruebas

### 3.6.1 Plan de Pruebas.

En esta sección se establecen los objetivos de calidad en el desarrollo del software, y se indican cuáles son las pruebas que permitirán determinar si el software posee un nivel de calidad aceptable en cuanto a exactitud, rendimiento, seguridad y usabilidad se refiere de acuerdo a lo establecido por el estándar ISO 9126.

#### 3.6.1.1 *Objetivos de calidad de software.*

- Exactitud: Cumplir con el 95% de casos de pruebas funcionales con resultados correctos.
- Rendimiento: Valores de latencia de sistema web menores o igual al promedio (2.7 segundos).
- Seguridad: Obtener 0 mensajes de alerta de alto nivel como respuesta del escaneo automático del sistema con la herramienta OWASP ZAP.

- Usabilidad: Cumplir con el 90% de casos de pruebas de forma satisfactoria

Los objetivos anteriormente mencionados son el resultado de la investigación realizada para validar y controlar formalmente la calidad del software, se tomaron como referencia los conceptos estudiados en el estándar ISO 9126. Dada la naturaleza del software desarrollado sólo se aplican pruebas de exactitud, rendimiento, seguridad y usabilidad.

### 3.6.2 Pruebas de exactitud.

Para realizar esta prueba se utilizaron las cartillas de casos de uso Tablas del 5 al 10 y casos de prueba en las Tablas del 11 al 16, se realizan las actividades correspondientes a cada caso y se verifica que el resultado sea el esperado de acuerdo con los requerimientos funcionales del sistema. Se crearon y analizaron nueve casos de pruebas funcionales, tanto para el atributo de seguridad como para el de exactitud, basándose en los casos de uso del sistema general y en la utilización de un escáner de código, lo cual nos permitió validar el correcto flujo y generación de información.

### Requerimientos de entorno

Servidor LINUX DEBIAN (BUSTER 10)

Servidor Apache

Python Versión 2.7.16

MongoDB 2.4.6

Owasp Zap 2.9

Hard Disk: 4 GB

### Casos de uso del sistema Web Scraping

Tabla 5: Caso de uso - Inicio de sesión

<b>Caso de uso</b>	Inicio de sesión	<b>Identificador</b>	001
<b>Referencias</b>	Ninguna	<b>Responsable</b>	Autores
<b>Actores</b>	Administrador		

<b>Precondición</b>	Obtener un usuario y contraseña válidos para el inicio de sesión.
<b>Post condición</b>	
<b>Criterio de éxito</b>	Ingreso y visualización del panel de usuario.
<b>Procedimiento</b>	Ingresar el usuario en el input USUARIO. Ingresar Contraseña en el input PASSWORD.
<b>Resultado esperado</b>	Generación de una sesión con los datos del usuario mediante una sesión web.

Tabla 6: Caso de uso - Generación JSON con URL específica

<b>Caso de uso</b>	Generación JSON con URL específica	<b>Identificador</b>	002
<b>Referencias</b>	Ninguna	<b>Responsable</b>	Autores
<b>Actores</b>	Administrador		
<b>Precondición</b>	Iniciar sesión en el sistema, Ingreso de una URL válida		
<b>Post condición</b>	Ninguna		
<b>Criterio de éxito</b>	Visualización en formato JSON de servicios existentes en la URL ingresada, el archivo JSON debe constar de los siguientes elementos (identificador, servicio, URL, información, importante, valores).		
<b>Procedimiento</b>	<ul style="list-style-type: none"> <li>● Inicio de sesión</li> <li>● Clic en el menú Web Scraping – submenú BUSCAR POR URL del panel interno del sistema.</li> <li>● Ingresar URL específica</li> <li>● Seleccionar tipo de búsqueda de Scraping – URL ESPECÍFICA</li> <li>● Clic en el botón BUSCAR</li> <li>● Guardar el resultado de la búsqueda en la base de datos MongoDB.</li> <li>● Visualización del JSON con los servicios extraídos de la URL ingresada.</li> </ul>		
<b>Resultado esperado</b>	<ul style="list-style-type: none"> <li>● Visualización completa de los servicios de la URL ingresada, en formato JSON.</li> </ul>		

Tabla 7: Caso de Uso - Generación JSON con dominio principal

<b>Caso de uso</b>	Generación JSON con dominio principal	<b>Identificador</b>	003
--------------------	---------------------------------------	----------------------	-----

<b>Referencias</b>	Ninguna	<b>Responsable</b>	Autores
<b>Actores</b>	Administrador		
<b>Precondición</b>	Iniciar sesión en el sistema, Ingreso de una URL válida		
<b>Post condición</b>	Ninguna		
<b>Criterio de éxito</b>	Visualización en formato JSON de servicios existentes en la URL ingresada, el archivo JSON debe constar de los siguientes elementos (identificador, servicio, URL, información, importante, valores).		
<b>Procedimiento</b>	<ul style="list-style-type: none"> <li>● Inicio de sesión</li> <li>● Clic en el menú Web Scraping – submenú BUSCAR POR URL del panel interno del sistema.</li> <li>● Ingresar dominio principal</li> <li>● Seleccionar tipo de búsqueda de Scraping – DOMINIO PRINCIPAL</li> <li>● Clic en el botón BUSCAR</li> <li>● Guardar el resultado de la búsqueda en la base de datos MONGODB.</li> <li>● Visualización del JSON con los servicios extraídos de la URL ingresada.</li> </ul>		
<b>Resultado esperado</b>	<ul style="list-style-type: none"> <li>● Visualización completa de los servicios de la URL ingresada, en formato JSON.</li> </ul>		

Tabla 8: Caso de uso - Generación de Reportes

<b>Caso de uso</b>	Generación de Reportes	<b>Identificador</b>	004
<b>Referencias</b>	002, 003	<b>Responsable</b>	Autores
<b>Actores</b>	Administrador		
<b>Precondición</b>	Iniciar sesión en el sistema		
<b>Post condición</b>	Ninguna		
<b>Criterio de éxito</b>	<p>Visualización tabla de historial de resultados, debe constar con las siguientes filas FECHA, HORA, USUARIO, URL, ESTADO, JSON.</p> <p>Al dar Clic en el botón VER de cada registro debe visualizar una ventana modal con los servicios encontrados, dispuestos en filas y columnas.</p>		

<b>Procedimiento</b>	<ul style="list-style-type: none"> <li>● Inicio de sesión</li> <li>● Clic en el menú REPORTE del panel interno del sistema.</li> <li>● Visualización mediante una tabla de todo el historial de resultados</li> <li>● Clic en el botón VER de cada registro de la tabla</li> <li>● Visualización del JSON con los servicios extraídos de la URL ingresada.</li> </ul>
<b>Resultado esperado</b>	Despliegue completo y correcto de registro de historiales de Resultados

Tabla 9: Caso de uso - Consumo de datos mediante servicio web

<b>Caso de uso</b>	Consumo de datos mediante servicio web	<b>Identificador</b>	005
<b>Referencias</b>	002,003	<b>Responsable</b>	Autores
<b>Actores</b>	Administrador		
<b>Precondición</b>	Iniciar sesión en el sistema		
<b>Post condición</b>	Ninguna		
<b>Criterio de éxito</b>	Visualización de JSON al seleccionar el historial de resultados.		
<b>Procedimiento</b>	<ul style="list-style-type: none"> <li>● Inicio de sesión</li> <li>● Clic en el menú WEBSERVICE del panel interno del sistema.</li> <li>● Seleccionar el historial de resultados que se desea visualizar.</li> <li>● Clic en el botón BUSCAR</li> <li>● Visualización del JSON con los servicios extraídos por el historial de resultados seleccionado.</li> </ul>		
<b>Resultado esperado</b>	Ver JSON con la información histórica completa de los servicios seleccionados por el historial de resultados.		

Tabla 10: Caso de uso - Historial de resultados

<b>Caso de uso</b>	Historial de resultados	<b>Identificador</b>	006
<b>Referencias</b>	002,003	<b>Responsable</b>	Autores
<b>Actores</b>	Administrador		
<b>Precondición</b>	Iniciar sesión en el sistema		

<b>Post condición</b>	Ninguna
<b>Criterio de éxito</b>	Despliegue mediante tablas de todos los historiales de resultados realizados de todos los usuarios, la tabla debe contener las siguientes columnas: FECHA, HORA, USUARIOS, URL, ESTADO Y JSON Debe constar opción de descarga del JSON en el modal anterior, mediante un botón de descarga.
<b>Procedimiento</b>	<ul style="list-style-type: none"> <li>● Inicio de sesión</li> <li>● Clic en el menú HISTÓRICOS del panel interno del sistema.</li> <li>● Visualización de la tabla de históricos de resultados con los servicios extraídos.</li> </ul>
<b>Resultado esperado</b>	Tabla donde se visualice de una manera ordenada todos los históricos de resultados que se han almacenada en la base de datos y también incluir un botón para ver el JSON por cada uno de los historiales.

En las (Tablas 5 hasta la 10) anteriormente expuestas se describen los diferentes casos de uso del sistema, en ellas se menciona el nombre del caso de uso, se le asigna a cada uno un número de identificación, se nombra a los responsables del desarrollo y al actor que podrá ejecutarlos, si existen pre y post condiciones, se explica el criterio de éxito, el procedimiento a ejecutar y el resultado esperado para cada caso. Cabe mencionar que estos casos de uso servirán como base para ejecutar los casos de prueba (Tablas 11 hasta la 16).

### Desarrollo de los casos de pruebas

Tabla 11: Caso de prueba Nro. 1

<b>Caso de Prueba</b>	Inicio de sesión	<b>Identificador</b>	CP_001
<b>Funcionalidad</b>	001	<b>Fecha</b>	21/06/2020
<b>Entradas</b>	Nombre de usuario Contraseña usuario		
<b>Salidas</b>	Ingreso a panel interno del sistema		
<b>Flujo de ejecución</b>	Se inició sesión con los siguientes datos: Nombre de usuario: ADMIN Contraseña: *****		

	Posteriormente redirige la página a la del perfil de usuario Que se encuentra en el panel interno del sistema.
<b>Procedimientos Especiales</b>	En caso de dejar vacíos los campos requeridos se generan una alerta al usuario para que ingrese los mismos. En caso de ingresar ya sea el usuario o contraseña, equivocada se indica al usuario un mensaje para que ingrese los datos correctos.
<b>Resultado</b>	<b>CORRECTO</b>

Tabla 12: Caso de prueba Nro.2

<b>Caso de Prueba</b>	Generación JSON con URL específica	<b>Identificador</b>	CP_002
<b>Funcionalidad</b>	002	<b>Fecha</b>	20/06/2020
<b>Entradas</b>	URL específica de sitio web que se va a analizar.		
<b>Salidas</b>	JSON con los siguientes apartados y la información de todos los servicios extraídos: identificador, servicio, información, importante y valores.		
<b>Flujo de ejecución</b>	URL: <a href="http://www.amt.gob.ec/index.php/servicios/matriculacion-vehicular-menu/cambio-de-servicio.HTML">http://www.amt.gob.ec/index.php/servicios/matriculacion-vehicular-menu/cambio-de-servicio.HTML</a> Seleccionar opción URL ESPECÍFICA Clic botón BUSCAR. Recarga la página web y se genera el JSON con toda la información de los servicios extraídos.		
<b>Procedimientos Especiales</b>	En caso de dejar vacíos los campos requeridos se da una alerta al usuario para que ingrese los mismos. En caso de ingresar una URL no válida se indica al usuario un mensaje para que ingrese los datos correctos.		
<b>Resultado</b>	<b>CORRECTO</b>		

Tabla 13: Caso de prueba Nro.3

<b>Caso de Prueba</b>	Generación JSON con dominio principal	<b>Identificador</b>	CP_003
<b>Funcionalidad</b>	003	<b>Fecha</b>	20/06/2020
<b>Entradas</b>	URL específica de sitio web que se va a analizar.		
<b>Salidas</b>	JSON con los siguientes apartados y la información de todos los servicios extraídos: identificador, servicio, URL, información, importante y valores.		

<b>Flujo de ejecución</b>	URL: <a href="http://www.amt.gob.ec/">http://www.amt.gob.ec/</a> Seleccionar opción DOMINIO PRINCIPAL Clic botón BUSCAR. Recarga la página web y se genera el JSON con toda la información de los servicios extraídos.
<b>Procedimientos Especiales</b>	En caso de dejar vacíos los campos requeridos, se genera una alerta al usuario para que ingrese los mismos. En caso de ingresar una URL no válida se indica al usuario un mensaje para que ingrese los datos correctos.
<b>Resultado</b>	<b>CORRECTO</b>

Tabla 14: Caso de prueba Nro.4

<b>Caso de Prueba</b>	Generación de reportes	<b>Identificador</b>	CP_004
<b>Funcionalidad</b>	004	<b>Fecha</b>	20/06/2020
<b>Entradas</b>	Seleccionar el JSON de resultados		
<b>Salidas</b>	Visualización tabla de historial de resultados, que consta con las siguientes filas: fecha, hora, usuario, URL, estado, JSON.		
<b>Flujo de ejecución</b>	Ingresar al menú Reportes Clic botón Ver. Visualización de la información del JSON seleccionado.		
<b>Procedimientos Especiales</b>	Seleccionar el JSON almacenado de acuerdo a la fecha.		
<b>Resultado</b>	<b>CORRECTO</b>		

Tabla 15: Caso de prueba Nro.5

<b>Caso de Prueba</b>	Consumo de datos mediante servicio web	<b>Identificador</b>	CP_005
<b>Funcionalidad</b>	005	<b>Fecha</b>	20/06/2020
<b>Entradas</b>	Seleccionar historial de resultados		
<b>Salidas</b>	JSON con los siguientes apartados y la información de todos los servicios extraídos: identificador, servicio, información, importante y valores.		
<b>Flujo de ejecución</b>	OPCIÓN: FECHA: 20/06/20 USUARIO: ADMIN Clic botón BUSCAR. Recarga la página web y se genera el JSON con toda la información de los servicios extraídos por el historial de resultados seleccionados.		



<b>Procedimientos Especiales</b>	En caso de dejar vacíos los campos requeridos se da una alerta al usuario para que ingrese los mismos.
<b>Resultado</b>	<b>CORRECTO</b>

Tabla 16: Caso de prueba Nro.6

<b>Caso de Prueba</b>	Historial de resultados	<b>Identificador</b>	CP_006
<b>Funcionalidad</b>	006	<b>Fecha</b>	20/06/2020
<b>Entradas</b>	Seleccionar el JSON de resultados		
<b>Salidas</b>	Visualización tabla de históricos que consta con las siguientes filas: fecha, hora, usuario, URL, estado, JSON.		
<b>Flujo de ejecución</b>	Ingresar al menú Históricos Clic botón Ver. Visualización de la información del JSON en el modal, que incluye la descarga mediante un botón de descarga.		
<b>Procedimientos Especiales</b>	Seleccionar el JSON almacenado de acuerdo a la fecha.		
<b>Resultado</b>	<b>CORRECTO</b>		

Tabla 17: Resultados desarrollo casos prueba

<b>Id caso de prueba</b>	<b>Funcionalidad</b>	<b>Fecha</b>	<b>Resultado correcto</b>
CP_001	Inicio de sesión	20/06/2020	SI
CP_002	Generación JSON con URL específica	20/06/2020	SI
CP_003	Generación JSON con dominio principal	20/06/2020	SI
CP_004	Generación de reportes	20/06/2020	SI
CP_005	Consumo de datos mediante servicio web	20/06/2020	SI
CP_006	Historial de resultados	20/06/2020	SI

Nota: En esta tabla se reflejan los resultados de las pruebas de funcionalidad según el caso de uso

En la Tabla 17, se muestra un resumen de los resultados obtenidos para cada caso de prueba y la fecha de ejecución de cada uno.

### **3.6.3 Pruebas de rendimiento.**

En este tipo de pruebas se somete al software a una carga de trabajo con el fin de medir su velocidad, fiabilidad y estabilidad, detectando cuellos de botella, localizando problemas de rendimiento y verificando el cumplimiento de los acuerdos de nivel de servicio. Para lo cual, se analizó las URL principales del sistema mediante casos de pruebas de rendimiento utilizando la herramienta web Octoperf, considerado como un software potente para evaluar el rendimiento de cualquier sistema informático. El proceso a aplicar, inicia sometiendo al software a la herramienta, luego, se realiza un testing con 50 usuarios (en un rango de edades de 25 a 48) de distintas localidades y bajo distintas características de arquitectura tecnológica (Laptop, Tablet, Móvil, etc.), donde proceden a manipular al mismo tiempo, cargando datos, archivos, entre otros. A continuación, se detalla resultados.

#### **Requerimientos de entorno**

Servidor LINUX DEBIAN (BUSTER 10)

Servidor Apache

Python Versión 2.7.16

MongoDB 2.4.6

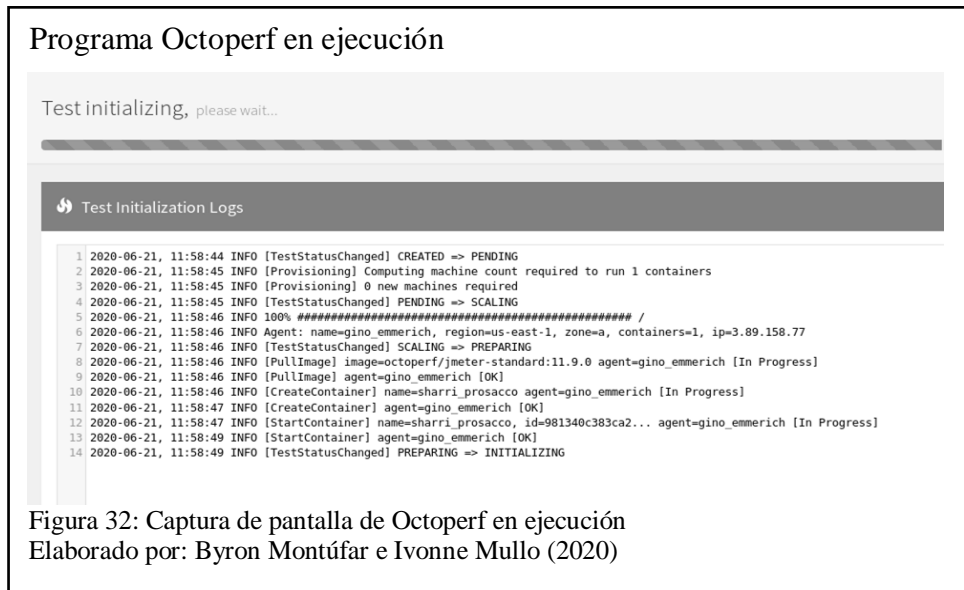
Hard Disk: 4 GB

#### **Herramientas**

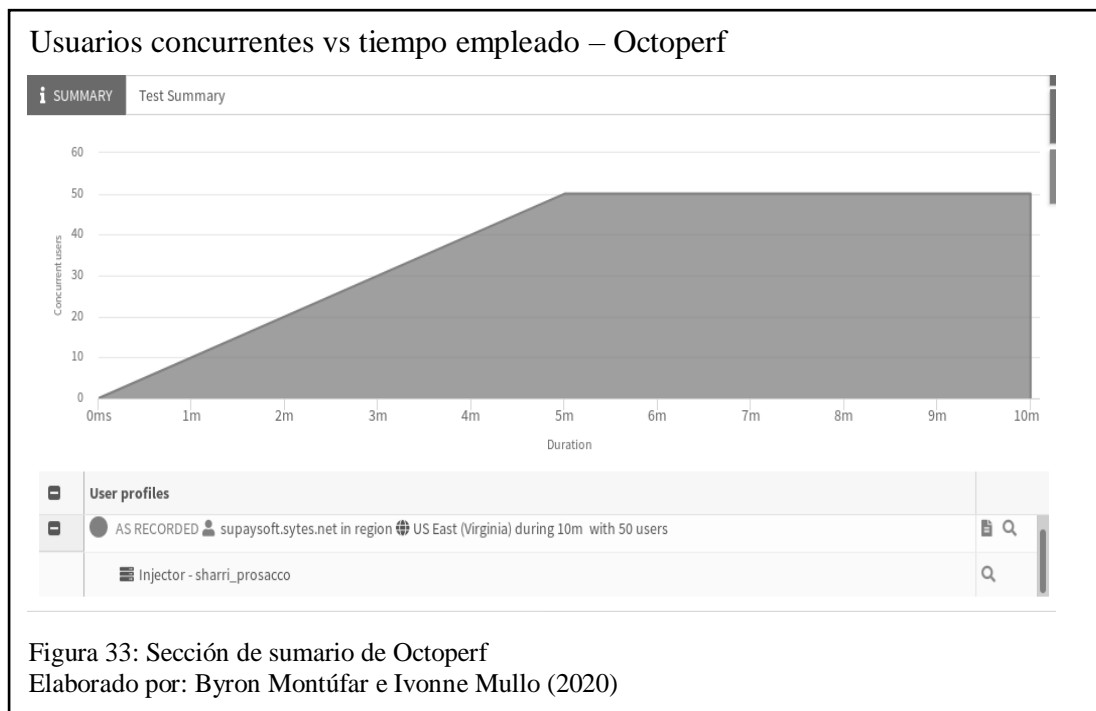
- Octoperf: “Herramienta basada en JMETER para pruebas de carga y de rendimiento de software, permite mediante una interfaz gráfica la creación de múltiples perfiles de usuarios virtuales para la ejecución de sistemas en varios escenarios” (OctoPerf.com ©, 2020). EL sistema en prueba deberá pasar por las siguientes fases DESING, RUNTIME, ANALYSIS para posteriormente obtener los resultados en gráficas estadísticas.
- Cartillas de casos de uso

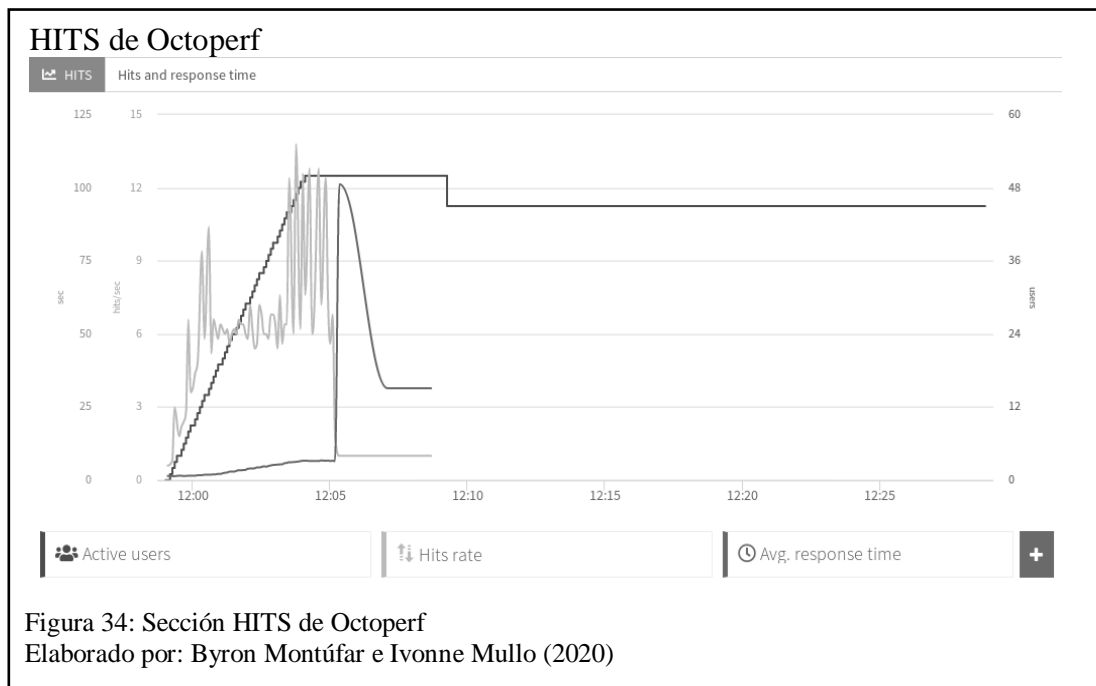
## Desarrollo pruebas de rendimiento

En la (Figura 32) se puede observar el test inicial para la verificación de URL principal y URL derivadas

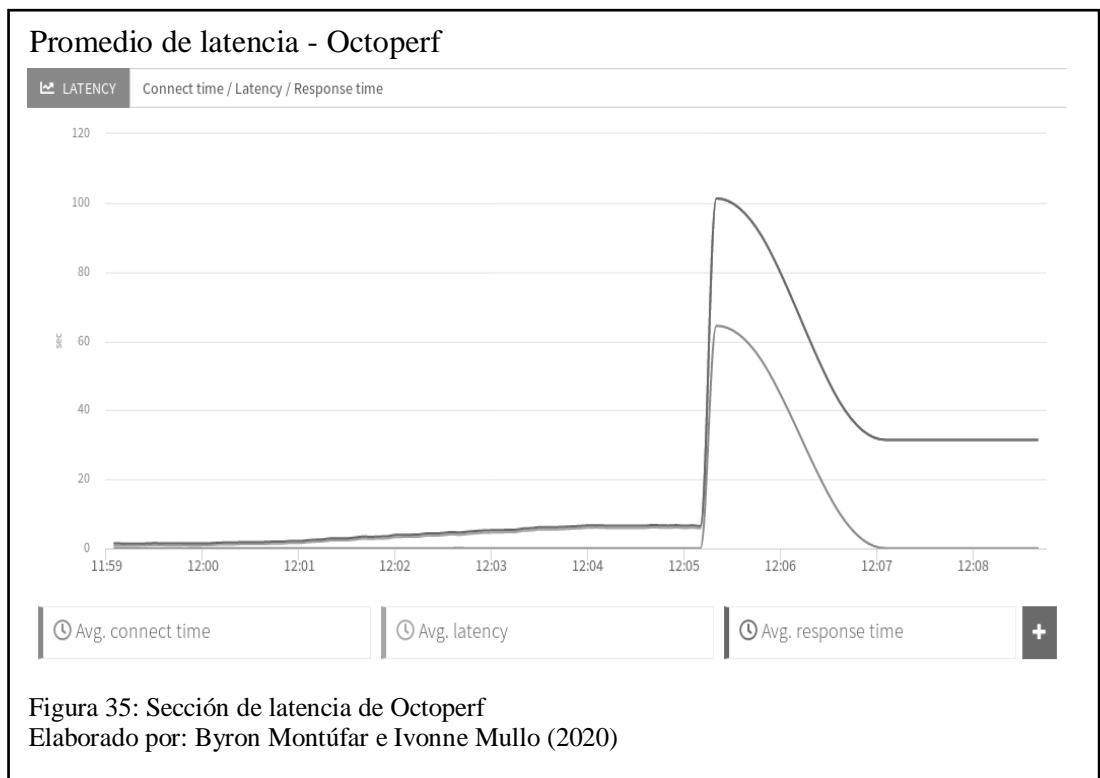


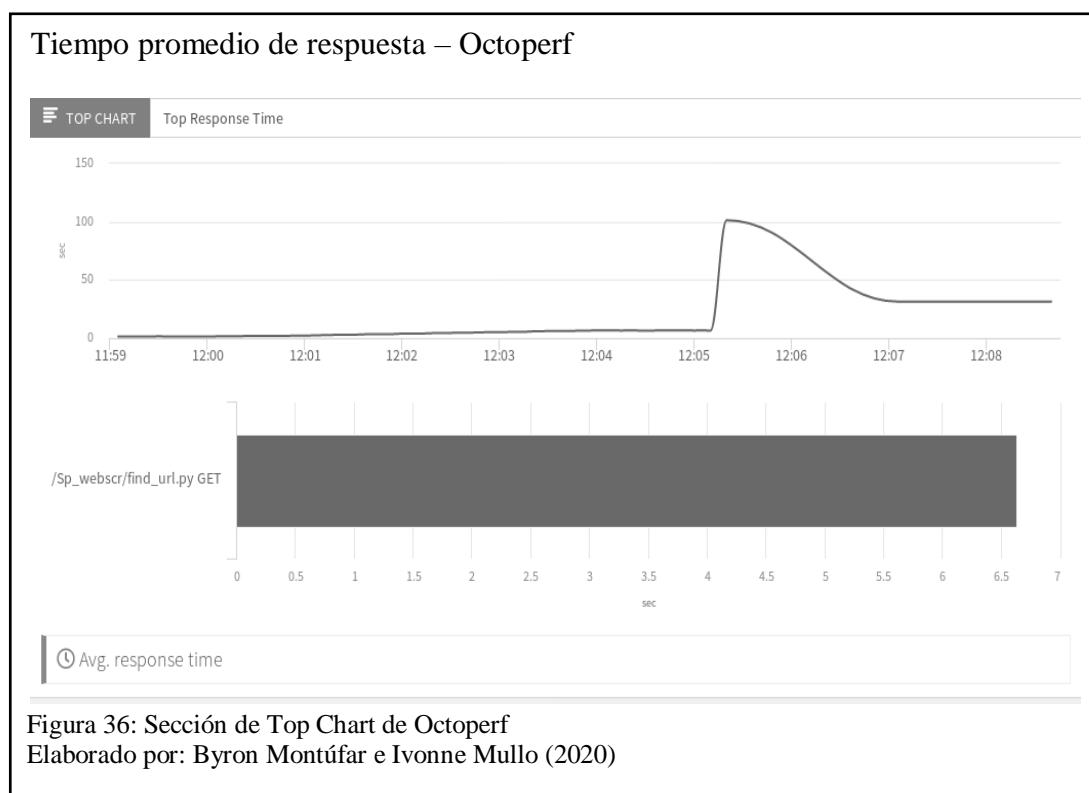
En la (Figura 33) se visualizan las estadísticas de la cantidad de usuarios concurrentes (conectados al mismo tiempo) en comparación con el tiempo empleado expresado en minutos.





En la (Figura 34) se muestra el promedio de las visualizaciones de una página o la transferencia de un archivo en períodos de tiempo de 5 minutos, mientras que en la (Figura 35) se puede visualizar el promedio de tiempo que tardan los datos en estar disponibles desde que se realiza su petición (latencia).





En la (Figura 36) se observa el resultado de la medición del promedio de tiempo de respuesta de las solicitudes realizadas al servidor mostrados de forma constante minuto a minuto.

### 3.6.4 Pruebas de seguridad.

Se refiere a la capacidad de evitar el acceso no autorizado, a las funciones y datos del sistema. En este proyecto se utilizó un escáner de seguridad web llamado OWASP ZAP. Debido a la naturaleza del sistema se aceptan hasta 10 alertas de niveles medio y bajo, sin embargo, no deben aparecer alertas de nivel alto al momento de finalizar el escaneo. Teniendo en cuenta que los niveles de alarmas están divididos según su riesgo y por el tipo de vulnerabilidad que presenta el sistema analizado, como se muestra en la Tabla 18.

Tabla 18: Descripción del nivel de alarma de seguridad

<b>Alarma / Riesgo</b>	<b>Vulnerabilidad</b>
Alto Nivel / Grave	XSS (Cross-Site Scripting) Redirecciones Externas Inclusión Archivos Remotos
Nivel Medio / Medio	Alteración de Parámetros
Nivel Bajo / Bajo	Cookies Header Tipo de Contenido Autocompletado de Contraseña Inclusión de Archivos JavaScript Protección XSS

Nota: en esta tabla se describen los niveles de alarmas según el tipo de vulnerabilidad. Adaptado de: (OWASP, 2019).

### **Ejecución de prueba de seguridad con OWASP ZAP**

En primer lugar, se ingresa la URL correspondiente a la interfaz de inicio de sesión del sistema y la aplicación mediante un crawling automático a partir de la URL, adquiere la lista de posibles request que puede realizar, posteriormente genera ataques que detecten vulnerabilidades, adicionalmente puede validar si la configuración de los headers http es correcta según el contenido de estos.

## Programa Owsap Zap en ejecución

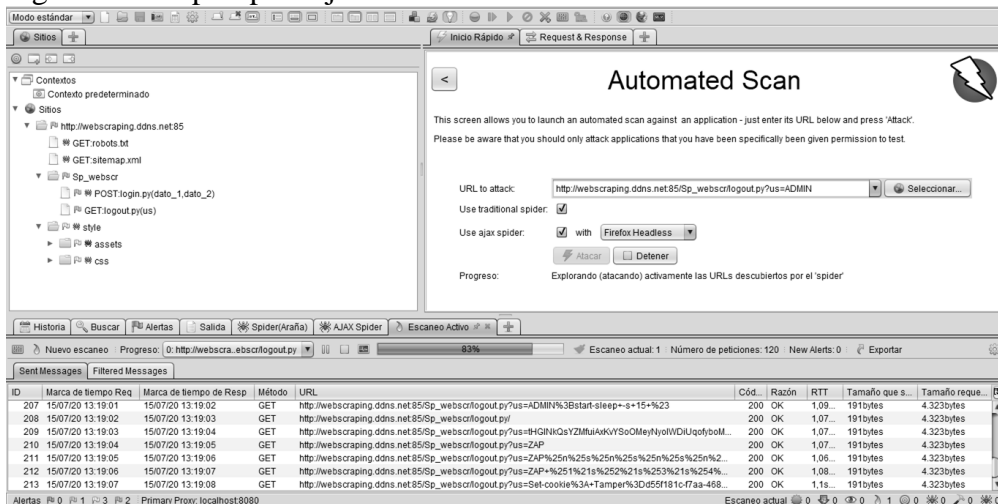


Figura 37: Owsap Zap en ejecución  
Elaborado por: Byron Montúfar e Ivonne Mullo (2020)

En la (Figura 37) se puede visualizar el programa Owsap Zap en ejecución en modo de escaneo automático. Luego en la (Figura 38) se genera el reporte de alertas de seguridad.

## Escaneo automático del sistema con Owsap Zap

### ZAP Scanning Report

#### Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	1
Low	3
Informational	2

#### Alert Detail

Medium (Medium)	Encabezado X-Frame-Options no establecido
Description	El encabezado X-Frame-Options no está incluido en la respuesta HTTP para proteger ante ataques 'ClickJacking'.
URL	http://webscraping.ddns.net:85/Sp_wbscrlogin.py
Method	POST
Parameter	X-Frame-Options
URL	http://webscraping.ddns.net:85/Sp_wbscrflogout.py?us=ADMIN
Method	GET
Parameter	X-Frame-Options
Instances	2
Solution	Los navegadores de web más modernos apoyan la cabecera HTTP X-Frame-Options. Asegúrese que está establecido en todas las páginas web devueltas por su sitio (si usted espera que la página esté enmarcada solo por páginas en su servidor (por ejemplo, es parte de un FRAMESET) entonces usted quemá usar SAMEORIGIN, de otra forma si usted nunca espera que la página esté enmarcada, debería usar DENY. ALLOW-FROM permite a sílica web específicas enmarcar la página web en navegadores web compatibles).
Reference	http://blogs.mdn.com/b/leinternals/archive/2010/03/30/combating-clickjacking-with-x-frame-options.aspx
CWE Id	16
WASC Id	15
Source ID	3
Low (Medium)	Inclusión de archivos de origen JavaScript Cross-Domain
Description	Las páginas incluyen uno o más archivos encriptados de un dominio de terceros.
URL	http://webscraping.ddns.net:85/Sp_wbscrflogout.py?us=ADMIN
Method	GET
Parameter	http://bx-code-admin.websitedesignmarketingagency.com/assets/vendor_components/sreenfull/screenfull.js
Evidencia	<script src="http://bx-code-admin.websitedesignmarketingagency.com/assets/vendor_components/sreenfull/screenfull.js"></script>
URL	http://webscraping.ddns.net:85/Sp_wbscrlogin.py
Method	POST
Parameter	http://bx-code-admin.websitedesignmarketingagency.com/assets/vendor_components/bootsrap/dist/js/bootsrap.min.js
Evidencia	<script src="http://bx-code-admin.websitedesignmarketingagency.com/assets/vendor_components/bootsrap/dist/js/bootsrap.min.js"></script>

Figura 38: Reporte de resultados de escaneo de Owsap Zap  
Elaborado por: Byron Montúfar e Ivonne Mullo (2020)

El resultado del escaneo arroja 1 alerta de nivel medio y 2 alertas de nivel bajo por lo que se cumple con el objetivo de calidad correspondiente a la seguridad del sistema.

En la (Figura 39) se puede visualizar el detalle de las alertas del programa Owasp Zap.

**Detalle de alertas Owasp Zap**

**Alerta de Nivel Medio obtenida en el escaneo de Owasp Zap:**

El encabezado X-Frame\_options no está incluido en la respuesta HTTP para proteger ante ataques 'ClicJacking'.

**Alertas de Nivel Bajo obtenidas en el escaneo de Owasp Zap:**

- Inclusión de archivos de origen JavaScript Cross-Domain.
- El encabezado Anti-MIME-Sniffing X-Content-Type-Options no estaba configurado para 'nosniff'.
- Una solicitud falsa entre sitios en un ataque que compromete y obliga a una víctima a enviar su solicitud HTTP a un destino objetivo sin su conocimiento o intención para poder realizar una acción como víctima.

Figura 39: Detalle de alertas Owasp Zap  
Elaborado por: Byron Montúfar e Ivonne Mullo (2020)

### **3.6.5 Pruebas de usabilidad.**

Para este tipo de pruebas como se muestra en la Tabla 19, se tomó en cuenta la norma ISO / IEC 9126-4 la cual “recomienda que las métricas de usabilidad deben incluir medidas de eficacia de usabilidad” (Figuroa, 2012, p. 2).

En este caso el proceso de testing inicia con seis casos de pruebas de usabilidad basándose en los casos de uso del sistema detallados con anterioridad, seguidamente se tomó como base a 10 usuarios (entre una edad de 25 – 45 años); quienes mediante la manipulación del sistema se fue analizando y evaluando, cabe mencionar, este proceso de prueba se aplicó utilizando herramientas de comunicación digital (zoom, Skype, meet, etc.) con el objetivo de observar el comportamiento de los usuarios, luego mediante una matriz se procede a recopilar los resultados del procedimiento manual



de cada caso de prueba de usabilidad, lo cual permitirá medir la eficacia, efectividad, facilidad de manejo y satisfacción del usuario con el producto de software.

### Requerimientos de entorno

Servidor LINUX DEBIAN (BUSTER 10)

MongoDB 2.4.6

Hard Disk: 4 GB

Python Versión 2.7.16

Servidor Apache

Tabla 19: Medición de eficacia

Actividad	Terminaron la tarea		Problemas encontrados
	Dentro del Tiempo	Fuera del tiempo	
Inicio de sesión	10	0	Ninguno
Generación de JSON por URL específica	10	0	Ninguno
Generación de JSON por dominio principal	10	0	Ninguno
Cambio de contraseña de usuario Administrador	10	1	1 usuario tuvo dificultad para colocar dos veces la misma contraseña
Visualización de resultados por historial	10	0	Ninguno
Visualización de Reportes	10	0	Ninguno

Nota: en esta tabla se realiza la medición de eficacia para varias actividades del sistema.

### 3.7 Perspectiva visual del sistema

La primera pantalla que se visualiza del sistema es la de inicio de sesión (Figura 39) y cuenta con dos casillas de llenado para que el usuario introduzca sus datos de acceso al sistema y un botón de “Ingresar” para realizar la validación de datos y redirigir al usuario a la interfaz siguiente.

## Pantalla de inicio de sesión del sistema

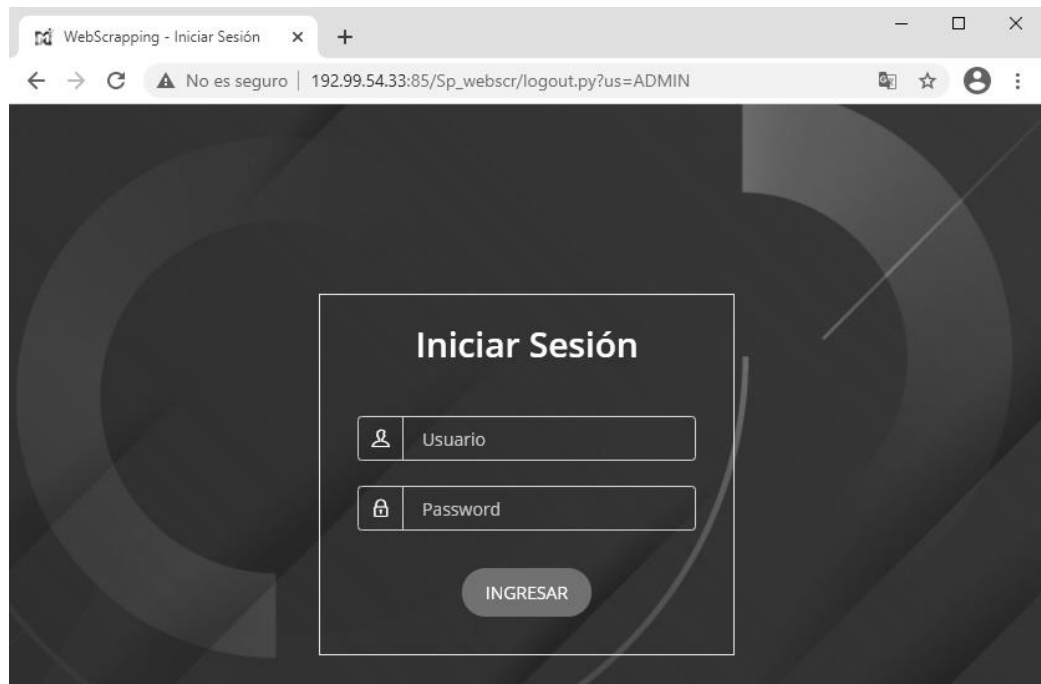


Figura 40: Interfaz para el inicio de sesión en el sistema  
Elaborado por: Byron Montúfar e Ivonne Mullo (2020)

Una vez validados los datos de acceso ingresados por el usuario se presenta la siguiente pantalla de la (Figura 40), donde se puede apreciar en el lateral izquierdo un menú principal, con las opciones: WebScrapping, WebServices, Históricos, Reportes. En el cuerpo de la interfaz se muestra el perfil del usuario, nombre y rol.

## Pantalla principal del Administrador

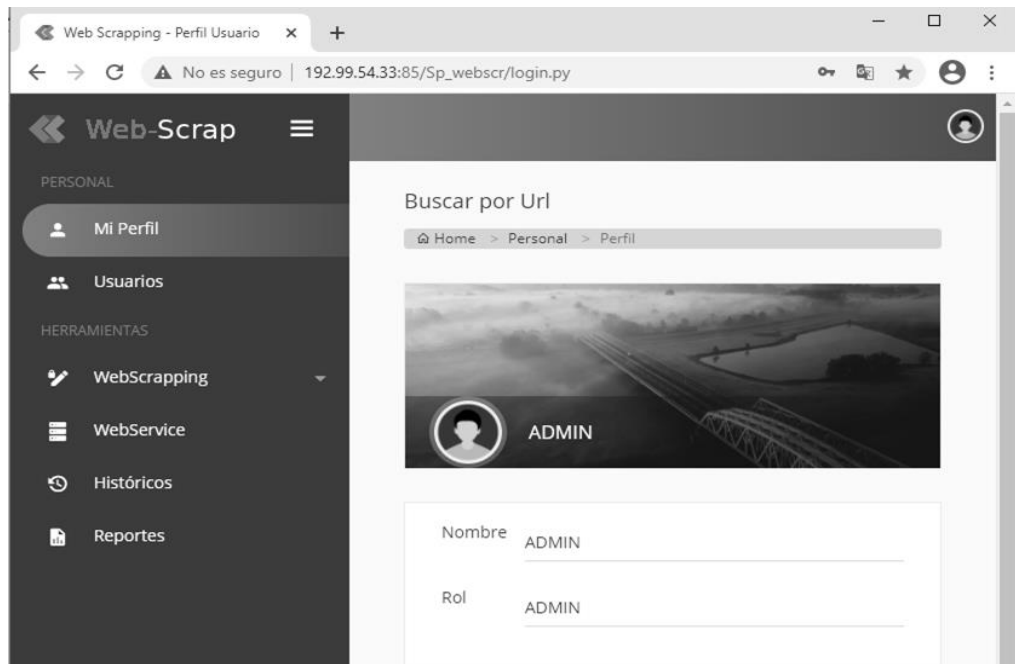


Figura 41: Interfaz principal del usuario Administrador  
Elaborado por: Byron Montúfar e Ivonne Mullo (2020)

La opción de webscraping (Figura 41), en esta opción permite realizar la extracción de la información de las páginas de la matriculación vehicular de las 3 principales ciudades, específicamente de la sección servicios.

## Opción del menú principal: webscraping

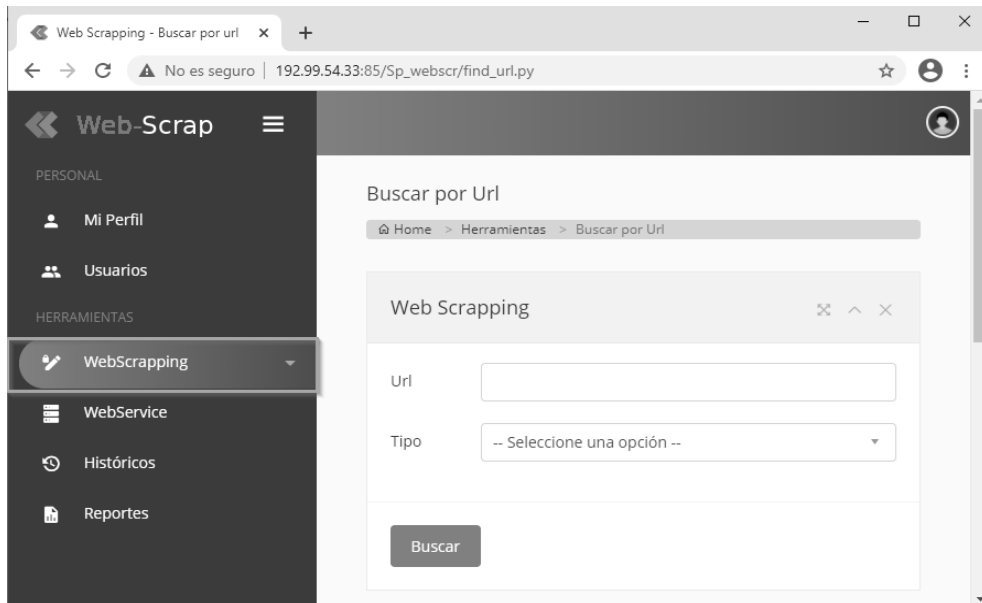


Figura 42: Opción del menú principal: webscraping  
Elaborado por: Byron Montúfar e Ivonne Mullo (2020)

La búsqueda se lo puede realizar de dos maneras: *Primer método de búsqueda:*

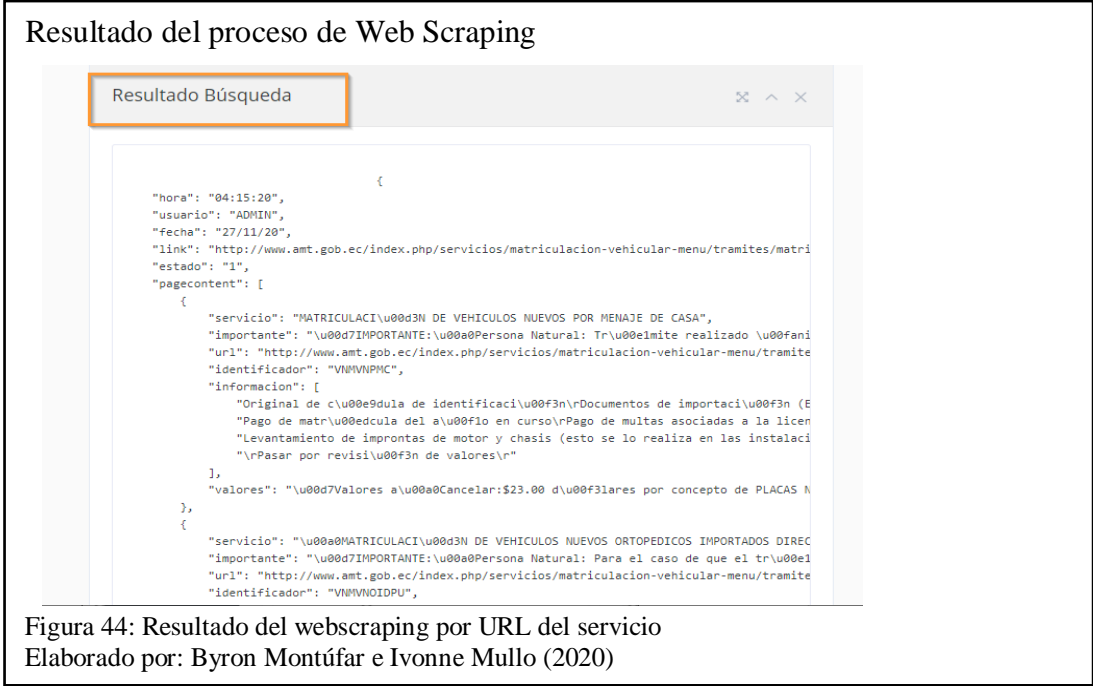
Ingresando la URL específica del servicio

- Se selecciona algún servicio a consultar y se copia la URL del servicio <http://www.amt.gob.ec/index.php/servicios/matriculacion-vehicular-menu/tramites/matriculacion.HTML>
- En el Web Scraping se debe pegar la URL del servicio y seleccionar la opción de tipo URL específica y presionar el botón Buscar. El ejemplo se puede visualizar en la (Figura 42).



Figura 43: Ejecución del webscrapin por url del servicio  
Elaborado por: Byron Montúfar e Ivonne Mullo (2020)

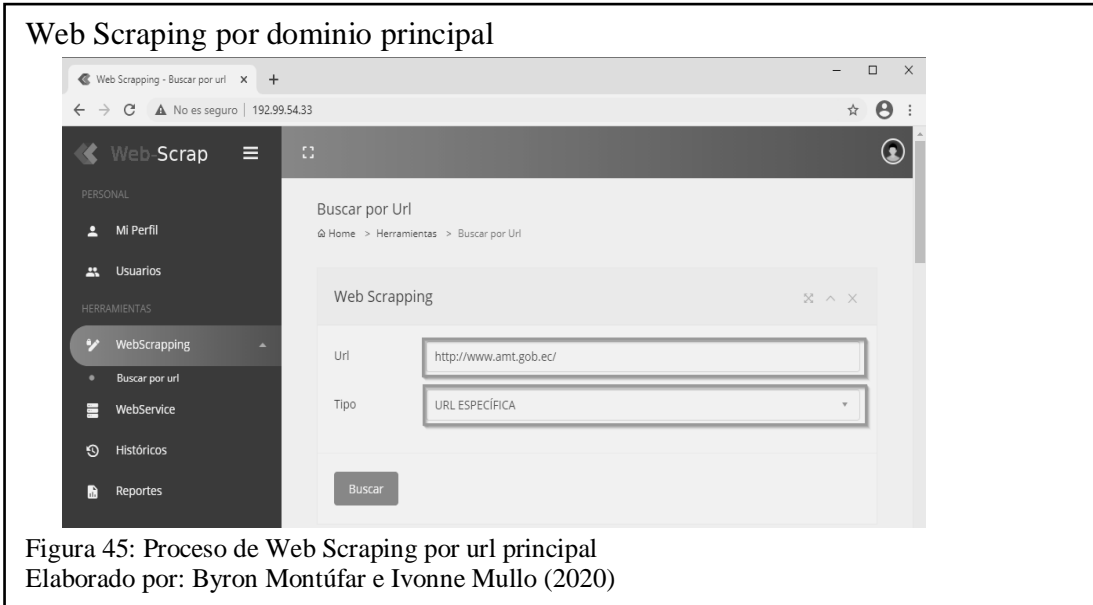
Y el resultado se visualiza en la sección de resultados de la misma interfaz (Figura 43).



Segundo método de búsqueda: Ingresando la URL del dominio principal:

- <http://www.amt.gob.ec/>
- <https://www.atm.gob.ec/>
- <https://www.emov.gob.ec/>

En el Web Scraping se debe pegar la URL del servicio, como se detalla en la (Figura 44), luego seleccionar la opción de tipo DOMINIO PRINCIPAL y presionar el botón BUSCAR. El resultado se visualizará en la sección de resultados de la misma interfaz.



## Opción Históricos

Muestra todas las consultas (Figura 45) que el usuario ha realizado, además el usuario puede ver y descargar el JSON que se creó.

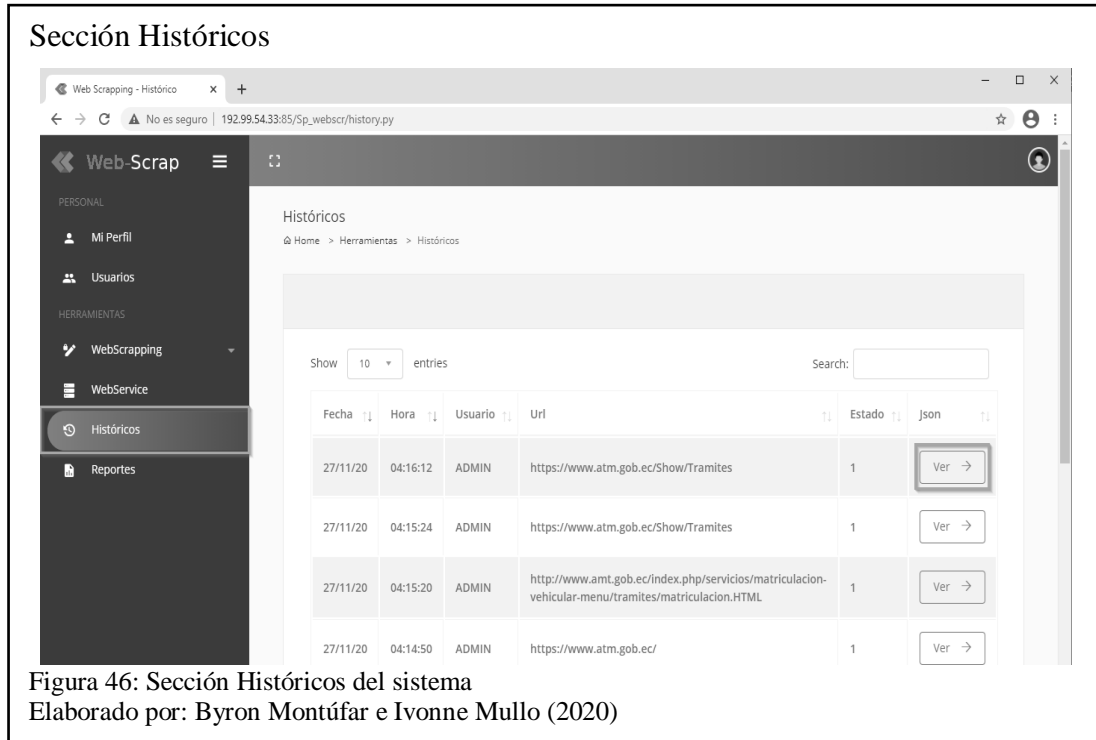


Figura 46: Sección Históricos del sistema  
Elaborado por: Byron Montúfar e Ivonne Mullo (2020)

Al momento de presionar en la opción Ver el usuario puede visualizar el archivo como se muestra en la (Figura 46) y descargar el JSON.

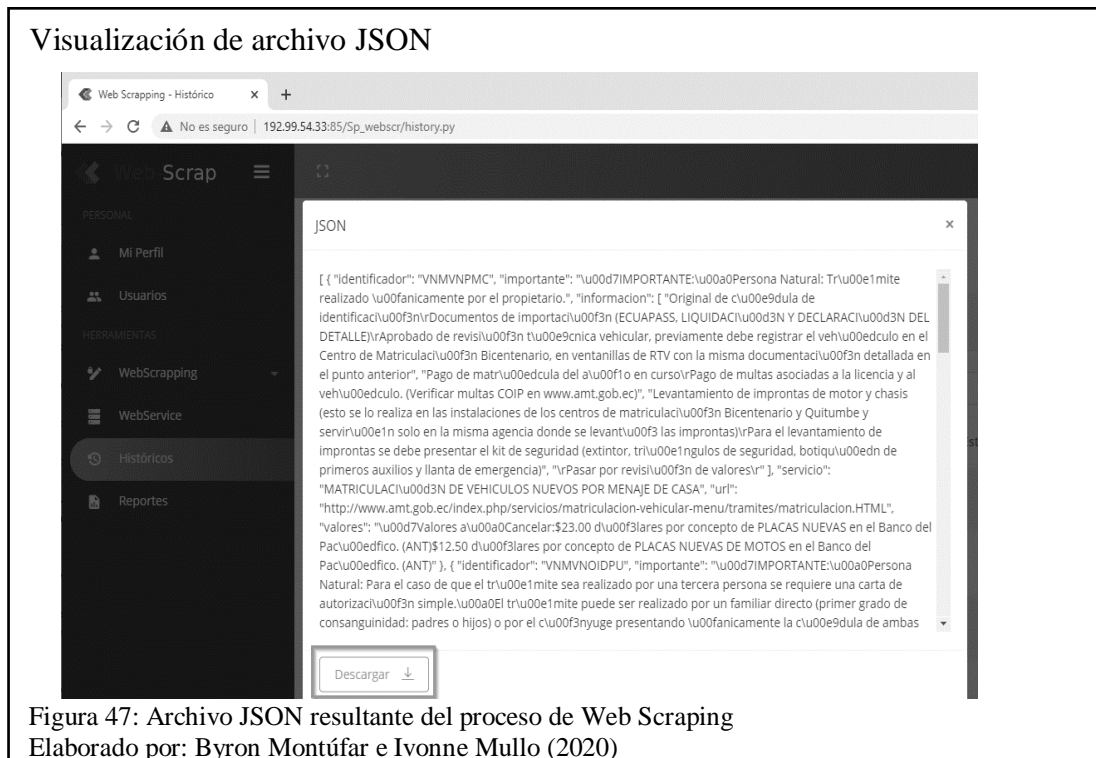


Figura 47: Archivo JSON resultante del proceso de Web Scraping  
Elaborado por: Byron Montúfar e Ivonne Mullo (2020)

Y el archivo que se descarga es presentado en la (Figura 47):

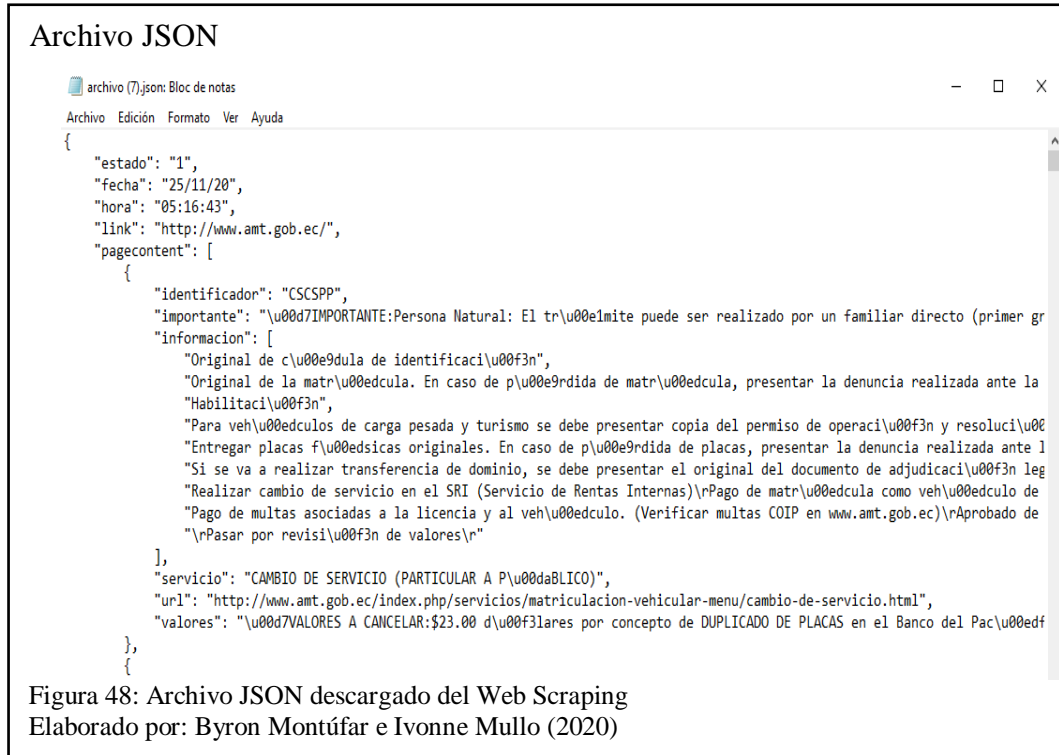


Figura 48: Archivo JSON descargado del Web Scraping  
Elaborado por: Byron Montúfar e Ivonne Mullo (2020)

### 3.8 Limitaciones y Restricciones

Durante la programación del algoritmo de Web Scraping se realizaron varias pruebas unitarias para verificar el funcionamiento adecuado de este. Para cada página objetivo (ANT, ATM y EMOV) se comparó la información extraída de forma automática y almacenada en el archivo JSON con la información mostrada en la página, esto se realizó para cada tipo de URL ingresada, principal y de servicios, en el (Anexo N° 09) se puede observar el archivo JSON generado con fecha 15/07/2020 donde se valida que la información extraída es la misma que se muestra en la página de la ATM (Guayaquil).

A nivel restrictivo, cabe mencionar que cuando se toman medidas adicionales para aumentar el nivel de seguridad de una página web como la instalación de un certificado de seguridad web, sea cual sea su tipo, este afectará directamente el proceso de extracción de datos. Según Soriano (2020) un certificado SSL “utiliza algoritmos

criptográficos y certificados de clave pública para verificar la identidad de las entidades que se comunican y para el intercambio de claves simétricas” (p. 64).

Esta autenticación puede generar problemas de conexión con respecto al algoritmo de Web Scraping ya que utilizan cifrado simétrico (criptografía simétrica) para ofrecer confidencialidad, y funciones de hash para asegurar la integridad de la información.

Las limitaciones que presenta el Web Scraping ante un mecanismo de seguridad como el utilizado por la ATM son sin duda motivo del protocolo de encriptación simétrica utilizado por la empresa certificadora, en el (Anexo N° 10) se puede apreciar el mensaje de error emitido al momento de ejecutar el Web Scraping en la URL principal de la ATM, ya que el Web Scraping no está programado para vulnerar la seguridad de dichos certificados, sólo se programó para realizar extracción de datos sin violar las normas de seguridad de dicha institución.



## Conclusiones

- En este trabajo de titulación se desarrolló una aplicación web para automatizar la extracción de la información de páginas web oficiales de las instituciones que ejercen la competencia de tránsito en las tres principales ciudades del país como lo son Quito, Cuenca y Guayaquil, (AMT, ATM y EMOV respectivamente), utilizando la técnica de Web Scraping para conocer los procesos de matriculación vehicular.
- El desarrollo de la aplicación logró solucionar la necesidad de alimentar de datos de entrada al Chatbot que desarrollan en el periodo 56 en la carrera de Ingeniería de Sistemas de la Universidad Politécnica Salesiana como proyecto de titulación y cuyo objetivo es ayudar a los usuarios acceder de una manera más sencilla y de forma precisa a la información acerca del proceso de matriculación vehicular en Quito, Guayaquil y Cuenca.
- Para lograr lo anteriormente descrito se investigó y analizó la utilidad, ventajas y desventajas de la técnica Web Scraping que se usó para la extracción de los datos. Se analizaron los requerimientos planteados por el ChatBot y se estructuraron los datos de acuerdo con lo requerido para la futura integración con los resultados del Web Scraping.
- Se codificó el algoritmo usando el lenguaje de programación Python, se diseñó la base de datos no relacional y la interfaz de usuario, también se probó la aplicación web con base en estándares de calidad de software como ISO/IEC 9126 y algunas definiciones realizadas por Pressman, Roger, 2010 en su libro titulado “Ingeniería del software. Un enfoque práctico”, logrando de esta manera la correcta extracción automática de información digital correspondiente a los procesos de matriculación vehicular.

- Se logró dentro del desarrollo del proyecto de titulación generar de forma satisfactoria el archivo JSON de acuerdo a las necesidades del ChatBot que será consumido en un futuro, para mostrar la información exacta de los procesos de matriculación vehicular en el menor tiempo posible.

## Recomendaciones

- El software desarrollado en este proyecto puede servir de base para el desarrollo de futuros trabajos, anexando los algoritmos para la extracción de datos de las páginas web de las entidades con competencias de tránsito que no fueron incluidas en el presente, como las ciudades de Ibarra, Ambato, Manta y Loja.
- Para la programación de un sistema como este, se recomienda usar siempre guía de estilo los estándares de codificación de Python ya que permiten al desarrollador obtener un código fuente ordenado, comprensible y sobre todo bien documentado.
- Se puede utilizar la investigación llevada a cabo en la fase de estudio de la técnica de Web Scraping como base referencial para futuros proyectos con objetivos similares.

## Referencias

- Alarcón, V. F. (2006). *Desarrollo de Sistemas de Información una Metodología Basada en el Modelado* (Primera ed.). Cataluña: EDICIONS UPC.
- AMT. (2020). *Agencia Metropolitana de Tránsito*. Obtenido de Base legal de creación de la AMT: <http://www.amt.gob.ec/index.php/quienes-somos/resolucion.html>
- ANT. (2020). *Agencia Nacional de Tránsito*. Obtenido de <https://www.ant.gob.ec/index.php/ant/vision-mision-y-objetivos>
- Appleton, Berczuk, & Cowham. (2007). Trazabilidad Lean-Agile: Estrategias y Soluciones. *CMCrossroads*, 10.
- Arias, F. G. (2006). *Proyecto de investigación: introducción a la metodología científica* (6 ta ed.). Caracas: Episteme. Obtenido de <http://ebevidencia.com/wp-content/uploads/2014/12/EL-PROYECTO-DE-INVESTIGACION-6ta-Ed.-FIDIAS-G.-ARIAS.pdf>
- ATM ®. (2020). *Autoridad de Tránsito Municipal*. Obtenido de ¿Quiénes somos?: <https://www.atm.gob.ec/Show/WhoWeAre>
- Brooke, J. (2012). *SUS-Aquickand dirtyusabilityscale*. Ucrania: Redhatch Consulting Ltd.
- Cuervo, Estupiñán, & Álvares. (2010). Heler: Una herramienta para la ingeniería de requisitos automatizada. *Unilibre Cali*, 6(2), 17.
- Diario EL TELÉGRAFO . (2014). *25 de 221 municipios han asumido las competencias de tránsito*. Obtenido de <https://www.eltelegrafo.com.ec/noticias/politica/3/25-de-221-municipios-han-asumido-las-competencias-de-transito-infografia>
- EMOV. (2020). *Empresa Pública Municipal de Movilidad, Tránsito y Transporte de Cuenca*. Obtenido de <https://www.emov.gob.ec/nosotros/>

- Fernandez, R. (2014). MongoDB: Qué Es, Cómo Funciona Y Cuándo Podemos Usarlo (o No). *Gembeta Dev*, 14.
- Figueroa, M. (2012). Calidad en la Industria del Software. La Norma ISO-9126. *Revista upiicsa*. Retrieved from <http://148.204.210.204/revistaupiicsa/34/34-2.pdf>
- Gavilánez, D., & Salas, J. (2016). *Bases de datos no relacionales: Utilización de Mongo DB como base de datos no relacional empleando formato GEO JSON*. Quito, Ecuador: UNIVERSIDAD POLITÉCNICA SALESIANA. doi:UPS - ST002829
- GAVIRIA, B. F. (2020). *PRUEBAS FUNCIONALES Y TÉCNICAS DE CAJANEGRA – PARTE I*. Cali - Colombia: Universidad del Valle. Obtenido de [https://campusvirtual.univalle.edu.co/moodle/pluginfile.php/1202320/mod\\_resource/content/1/2017A\\_CPS\\_CajaNegraParticionEquivalenciaValorLimite.pdf](https://campusvirtual.univalle.edu.co/moodle/pluginfile.php/1202320/mod_resource/content/1/2017A_CPS_CajaNegraParticionEquivalenciaValorLimite.pdf)
- Giménez, J. (2020). *ANÁLISIS Y VERIFICACIÓN DE UN SISTEMA DE CONTROL DE SEGURIDAD EN DISPOSITIVOS ELECTRÓNICOS*. Valencia, España: Escuela Técnica Superior de Ingeniería de Telecomunicación - Universitat Politècnica de València. doi:www.etsit.upv.es
- IEBS. (2019). Las metodologías ágiles más utilizadas y sus ventajas dentro de la empresa. (V. R. VILLÁN, Ed.) *Innovation & Entrepreneurship Business School*, 9.
- Izaurrealde, & Andriano. (2015). Modelo Adaptable de Trazabilidad de Requerimientos de Software en Entornos Ágiles de gran escala. In P. Izaurrealde, & N.

- Andriano, *Trazabilidad Ágil* (p. 12). Buenos Aires, Argentina: Universidad Tecnológica Nacional, Facultad Regional Córdoba.
- Jaramillo, J., & Sainz, L. (2012). *JLOP - (JSON LANGUAGE ORIENTED PROCESSING)*. Madrid: Universidad Complutense Madrid.
- Márquez, Sampedro, & Vargas. (2002). Instalación y configuración de Apache, un servidor web gratis. *Revista Científica Ingeniería y Desarrollo*(12), 14.
- Martí, M. (2016). Qué es el Web scraping? Introducción y herramientas. *SiteLabs*, 7.
- Molina, Loja, Zea, & Loaiza. (2016). Evaluación de los Frameworks en el Desarrollo de Aplicaciones Web con Python. *Latinoamericana de Ingeniería de Software*, 4(4), 201-207. Retrieved from <http://revistas.unla.edu.ar/software/article/view/1149>
- OctoPerf.com ©. (2020). *api.octoperf.com*. Obtenido de <https://api.octoperf.com/app/>
- Octopus Data Inc. (2020). ¿Qué es el Web Scraping y Por Qué es Importante? *Octoparce*, 5.
- OWASP. (2019). *OWASP Zed Attack Proxy Project*. Obtenido de Introduction: <https://www.zaproxy.org/docs/api/#introduction>
- Pressman, R. (2010). *INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO* (Séptima ed.). D. F., México: MCGRAW-HILL INTERAMERICANA EDITORES, S.A. DE C.V.
- Project Management Institute, Inc. (2017). *La guía de los fundamentos para la dirección de proyectos (Guía del PMBOK)*. Pennsylvania 19073-3299 EE.UU.: Project Management Institute, Inc.
- Soriano, M. (2020). *Seguridad en redes y seguridad de la información* (1era ed.). Praha, Czech Republic : Innovative Methodology for Promising VET Areas.

- Tabares, M., Barrera, A., Arroyave, J., & Pineda, J. (2007). Un método para la trazabilidad de requisitos en el Proceso Unificado de Desarrollo. *Revista EIA*, 69-82.
- Tamayo, T. y. (2003). *El Proceso de la Investigación Científica* (4 ta ed.). Mexico: Limusa Noriega Editores.
- Testeandosoftware ©. (2016). *OctoPerf. Pruebas de rendimiento realistas y accesibles*. Obtenido de <https://testeandosoftware.com/octoperf-pruebas-de-rendimiento-realistas-y-accesibles/>
- Testing Baires ©. (2016). *Performance Testing y una introducción al OctoPerf*. Obtenido de <https://testingbaires.com/2017/01/05/performance-testing-una-introduccion-al-octoperf/>
- Torrente, M. d. (2011). *sistema de evaluación de la usabilidad web orientado al usuario y basado en la determinación de tareas críticas*. España: Universidad de Oviedo.
- Varsovia, b., Hylton, J., Goodger, D., & Coghlan, N. (2020). Propósito y pautas de PEP. En P. S. Foundation, *Guía del desarrollador de Python* (pág. 96). python.org.
- www.pmoinformatica.com. (2018). *Pruebas de caja negra: Ejemplos* . Obtenido de <http://www.pmoinformatica.com/2017/02/pruebas-de-caja-negra-ejemplos.html>

