



**UNIVERSIDAD CATÓLICA
DE LA SANTÍSIMA CONCEPCIÓN
FACULTAD DE INGENIERÍA**

DESARROLLO DE UNA APLICACIÓN PARA
BÚSQUEDA Y EXTRACCIÓN DE TEXTO
DESDE LA WEB PARA EL ANÁLISIS DE LOS
SENTIMIENTOS

Por

LUIS ANDRÉS CORREA FERNÁNDEZ

INFORME DE PROYECTO DE TÍTULO APLICADO PARA
OPTAR AL TÍTULO DE
INGENIERO CIVIL INFORMÁTICO

Profesor guía:
CLAUDIA MARTÍNEZ ARANEDA

Comisión informante:
JOSE ABREU SALAS

Concepción, Chile
2018

Resumen

El Procesamiento Del Lenguaje Natural (PLN) es una disciplina de la Inteligencia Artificial que se ocupa de la formulación e investigación de mecanismos computacionales para la comunicación entre hombre-máquina mediante el uso de Lenguajes Naturales que pueden ser escritos, hablados y gestuales.

Una arista del PLN es el Análisis de los Sentimientos, esta se puede descomponer en dos enfoques, el primero es el análisis afectivo, donde se desarrollan y aplican técnicas para estimar el aspecto emotivo de un texto en particular basados en teorías psicológicas. El segundo es el análisis de polaridad donde se estima el nivel de polaridad presente en los textos, estos pueden ser clasificados en 4 niveles, positivo, negativo, neutro y ambiguo.

El presente proyecto tiene por objeto el desarrollo de una aplicación extractora de datos desde la Web y APIs públicas para apoyar en la fase inicial del proceso de análisis de los sentimientos correspondiente a la búsqueda y extracción de datos, esta aplicación se basa en el concepto de Scraping.

La aplicación cuenta con 3 servicios, un servicio de extracción de textos desde Twitter mediante el ingreso de una cadena de búsqueda, un servicio de creación de extractores personalizando sus características para extraer textos desde páginas web específicas y un servicio de almacenamiento de extractores para volver a utilizarlos.

Durante el transcurso de este documento se muestran distintas etapas del proyecto, desde bibliografía y definiciones de conceptos generales, hasta las etapas de desarrollo de la aplicación Text Extraction Community (TECO).

Índice general

1. Introducción	2
1.1. Contexto	2
1.2. Objetivo General	3
1.3. Objetivos Específicos	3
1.4. Metodología	4
1.5. Justificación	6
1.6. Limitaciones	7
1.7. Paradigma de Desarrollo de Software	8
1.8. Carta Gantt	9
2. Desarrollo	10
2.1. Conceptos generales sobre Procesamiento de Lenguaje Natural y Análisis de los Sentimientos.	10
2.1.1. Procesamiento Lenguaje Natural (PLN).	10
2.1.2. Análisis de los Sentimientos (AS).	12
2.1.3. Glosario de Términos Relevantes.	14
2.2. Herramientas con propósitos similares (Estado del Arte).	16
2.2.1. Import.io	16
2.2.2. ParseHub	18
2.2.3. Octoparse	20
2.2.4. Scraper Google Extension	22
2.3. Herramientas de programación	24
2.3.1. React.js.	24
2.3.2. Node.js.	26

2.3.3.	MongoDB.	28
2.3.4.	GitHub.	29
2.4.	Desarrollo de la aplicación Text Extraction Community	30
2.4.1.	Primer Incremento	31
2.4.1.1.	Especificación de Requisitos	34
2.4.1.1.1.	Requisitos Funcionales	34
2.4.1.1.2.	Requisitos no Funcionales	49
2.4.1.2.	Diseño de Interfaz y Modelo de Datos	50
2.4.1.2.1.	Mockups	50
2.4.1.2.2.	Diagrama de Actividades	56
2.4.1.3.	Codificación	58
2.4.1.4.	Validación y Verificación	67
2.4.2.	Segundo Incremento	77
2.4.2.1.	Especificación de Requisitos	79
2.4.2.1.1.	Requisitos Funcionales	79
2.4.2.1.2.	Requisitos no Funcionales	91
2.4.2.2.	Diseño de Interfaz y Modelo de Datos	92
2.4.2.2.1.	Mockups	92
2.4.2.2.2.	Diagrama de Actividades	98
2.4.2.2.3.	Diagrama de Despliegue	103
2.4.2.2.4.	Modelamiento de Datos	104
2.4.2.3.	Codificación	107
2.4.2.4.	Validación y Verificación	118
3.	Conclusiones	125
	Referencias	128
	Apéndices	132
A.	Manual de Usuario	132
A.1.	Proceso de Instalación	132
A.2.	Interfaces de la Aplicación TECO	134

Índice de figuras

2.1. Ejemplo: Herramienta Import.io	17
2.2. Ejemplo: Herramienta ParseHub	18
2.3. Ejemplo: Herramienta Octoparse	20
2.4. Ejemplo: Herramienta Scraper Google Extension	22
2.5. React Component.	26
2.6. Diagrama de Programación Orientada a Eventos.	27
2.7. Ejemplo: Consulta find() MongoDB.	29
2.8. Interfaz Extractor.	51
2.9. Interfaz Crear Nodo Extractor.	52
2.10. Interfaz Guardar Nodo Extractor.	53
2.11. Interfaz Editar Nodo Extractor.	54
2.12. Interfaz Guardar Extractor.	55
2.13. Diagrama de Actividades Extractor Personalizado.	57
2.14. State del Componente Extractor Personalizado.	62
2.15. Función Eliminar Nodo Extractor del Componente Extractor Personalizado.	63
2.16. Función Visualización Resultados Probar Nodo del Componente Extractor Personalizado.	64
2.17. Función Render del Componente Extractor Personalizado.	65
2.18. Export del Componente Extractor Personalizado.	66
2.19. Interfaz Extractor de Twitter.	93
2.20. Interfaz Opciones de Formato.	94
2.21. Interfaz Visualización de Resultados.	95
2.22. Interfaz Mis Extractores.	96

2.23. Interfaz Editar Extractor.	97
2.24. Diagrama de Actividades Inicio de la Plataforma.	99
2.25. Diagrama de Actividades Extractor de Twitter.	100
2.26. Diagrama de Actividades Mis Extractores.	102
2.27. Diagrama de Despliegue Aplicación TECO.	103
2.28. Especificación JSON aplicación TECO.	105
2.29. Adaptación del JSON a Entidad-Relación.	106
2.30. State del Componente Extractor Twitter.	109
2.31. Componente interno Modal de Tweets Extraídos del Componente Extractor Twitter.	110
2.32. Función Render del Componente Extractor Twitter.	111
2.33. Export del Componente Extractor Twitter.	112
2.34. State del Componente Mis Extractores.	113
2.35. Funciones del Componente Mis Extractores.	114
2.36. Función Render del Componente Mis Extractores.	115
2.37. Export del Componente Mis Extractores.	116
2.38. End-Points del Servidor de la Aplicación TECO.	117
3.1. Ejecutar MongoDB	132
3.2. Inicio de MongoDB en puerto 27017 por default	132
3.3. Instalar Yarn vía Node Package Manager	133
3.4. Carpetas de la Aplicación TECO	133
3.5. Iniciar Interfaz Web de la Aplicación TECO	133
3.6. Iniciar Servidor de la Aplicación TECO	133
3.7. Página inicio de la aplicación TECO.	134
3.8. Información reducida del extractor de twitter de la página de inicio.134	
3.9. Información reducida del extractor personalizado de la página de inicio.	135
3.10. Página de información formatos de descargas.	135
3.11. Página registro de nuevo usuario a la plataforma.	136
3.12. Página ingreso de usuario a la plataforma.	137

3.13. Página ingreso de datos personales.	138
3.14. Página ingreso de palabra clave.	138
3.15. Página ingreso de nueva contraseña.	138
3.16. Página inicio usuario con ingreso exitoso en la plataforma.	139
3.17. Página de selección de servicio.	139
3.18. Página información cómo usar extractor twitter.	140
3.19. Página operadores de búsqueda para extractor de twitter.	141
3.20. Página del extractor de twitter.	141
3.21. Página de ejemplo del extractor de twitter.	142
3.22. Página de visualización de tweets obtenidos.	142
3.23. Página de selección de formato de descarga de tweets obtenidos.	142
3.24. Página información cómo usar extractor personalizado.	143
3.25. Página terminología en la que se basa el extractor personalizado.	144
3.26. Página de recomendaciones para el usuario.	144
3.27. Página de creación del extractor personalizado.	145
3.28. Página para creación de nodos extractores.	145
3.29. Página de ejemplo de creación y prueba de extractor.	145
3.30. Página de visualización para los textos extraídos al probar el extractor.	146
3.31. Página de visualización para los textos extraídos al probar nodos.	146
3.32. Página para selección de formato de descargas.	146
3.33. Página de listado de extractores del usuario.	147
3.34. Página de visualización de las características de un extractor seleccionado.	148

Índice de tablas

2.1. Test Unitario para función Crear Elemento Extractor.	67
2.2. Test Unitario para función Eliminar Elemento Extractor.	68
2.3. Test Unitario para función Crear Nodo Extractor.	68
2.4. Test Unitario para función Probar Nodo Extractor.	69
2.5. Test Unitario para función Editar Nodo Extractor.	70
2.6. Test Unitario para función Guardar Nodo Extractor.	71
2.7. Test Unitario para función Eliminar Nodo Extractor.	72
2.8. Test Unitario para función Crear Extractor Personalizado.	72
2.9. Test Unitario para función Guardar Extractor Personalizado.	73
2.10. Test Unitario para función Probar Extractor Personalizado.	74
2.11. Test Unitario para función Descargar Datos Extraídos por Nodos.	75
2.12. Test Unitario para función Visualizar Datos Extraídos por Nodos.	75
2.13. Test Unitario para función Descargar Datos Extraídos por Extractores.	76
2.14. Test Unitario para función Visualizar Datos Extraídos por Extractores.	76
2.15. Test Unitario para función Buscar Tweets.	118
2.16. Test Unitario para función Descargar Datos Extraídos por Extractores.	119
2.17. Test Unitario para función Visualizar Tweets Extraídos.	119
2.18. Test Unitario para función Cargar Mis Extractores.	120
2.19. Test Unitario para función Eliminar Extractor Personalizado.	120
2.20. Test Unitario para función Editar Extractor Personalizado.	121
2.21. Test Unitario para función Registrar Usuario.	122

2.22. Test Unitario para función Ingresar Usuario.	123
2.23. Test Unitario para función Cambiar Contraseña.	124

Capítulo 1

Introducción

1.1. Contexto

El grupo de investigación SOMOS (Software Modelling Science), conformado por académicos de la Universidad del Bío-Bío y la Universidad Católica de la Santísima Concepción, posee entre sus investigaciones el análisis de los sentimientos, para cumplir su propósito definieron la siguiente metodología:

- Exploración, Extracción.
- Pre-procesamiento.
- Procesamiento.
- Análisis y Evaluación.

Para la investigación, desean analizar textos digitales de portales chilenos, como por ejemplo, diarios digitales, foros, Twitter, Facebook, entre otros, es por esta razón que desearían contar con una herramienta tecnológica que les permita extraer información desde los portales para reducir el esfuerzo en términos de tiempo comparado con la estrategia de realizar la extracción de forma manual.

1.2. Objetivo General

Desarrollar una aplicación extractora de textos desde páginas web y APIs públicas basada en criterios de búsqueda

1.3. Objetivos Específicos

Para lograr el objetivo general se definió la siguiente metodología de trabajo:

1. Estudiar conceptos generales sobre Procesamiento de Lenguaje Natural y Análisis de los Sentimientos.
2. Estudiar sobre herramientas con propósitos similares (Estado del Arte).
3. Estudiar herramientas de programación.
4. Desarrollar aplicación Text Extraction Community (TECO).

1.4. Metodología

Para lograr cada objetivo específico :

- **Para el objetivo específico 1:**

- 1.1 Revisar bibliografía de artículos relacionados con las palabras claves Procesamiento de Lenguaje Natural, Análisis de los Sentimientos, entre otras.

- 1.2 Construir glosario de términos relevantes.

- **Para el objetivo específico 2:**

- 2.1 Explorar Internet en busca de aplicaciones de extracción de textos.

- 2.2 Rescatar y evaluar ideas encontradas.

- 2.3 Generar propuesta solución.

- **Para el objetivo específico 3:**

- 3.1 Estudiar React.js.

- 3.2 Estudiar Node.js.

- 3.3 Estudiar MongoDB.

- 3.4 Estudiar GitHub.

■ **Para el objetivo específico 4:**

4.1 Etapa de Planificación.

4.1.1 Seleccionar paradigma de Desarrollo de Software.

4.2.2 Generar perfil del proyecto.

4.3.3 Generar carta Gantt.

4.2 Etapa de Análisis.

4.2.1 Capturar requisitos del usuario.

4.2.2 Definir requisitos finales funcionales.

4.2.3 Definir requisitos finales no funcionales.

4.2.4 Generar documentación Especificación de Requisitos.

4.3 Etapa de Diseño.

4.3.1 Definir modelo de flujo de datos.

4.3.2 Definir interfaz de usuario.

4.3.3 Generar Esquemas de la Base de Datos.

4.4 Etapa de Codificación.

4.4.1 Crear funciones modulares.

4.4.2 Generar comentarios.

4.4.3 Acoplar funcionalidades.

4.5 Etapa de Pruebas.

4.5.1 Verificar aplicación.

4.5.2 Corregir errores.

4.5.3 Validar aplicación.

1.5. Justificación

- Este proyecto aportará al trabajo del Grupo de investigación SOMOS (Software Modelling Science, código GI 170315/EF), que incorpora académicos de la Facultad de Ciencias Empresariales de la Universidad del Bío-Bío y del Departamento de Ingeniería Informática de la UCSC, que requieren una herramienta tecnológica personalizada para cubrir sus necesidades.
- La etapa inicial del proceso de Análisis de Sentimientos (Extracción de Datos) implica esfuerzos en términos de tiempo de extracción, por lo que el desarrollo de esta aplicación sería un aporte a esta fase.
- En la actualidad existen herramientas que realizan la tarea de extracción de datos, pero generalmente son herramientas de pago y las que son gratuitas no alcanzan a cubrir las necesidades del grupo de investigación.

1.6. Limitaciones

- **Maquetación:**

Las páginas web no necesariamente poseen la misma estructura debido a que dependen del estilo de programación del programador.

- **Alta Complejidad:**

La creación de extractores genéricos conlleva una alta complejidad.

- **Anti-Extractores:**

Existen páginas web que cambian el posicionamiento de sus elementos constantemente, lo cual dificulta el accionar del extractor.

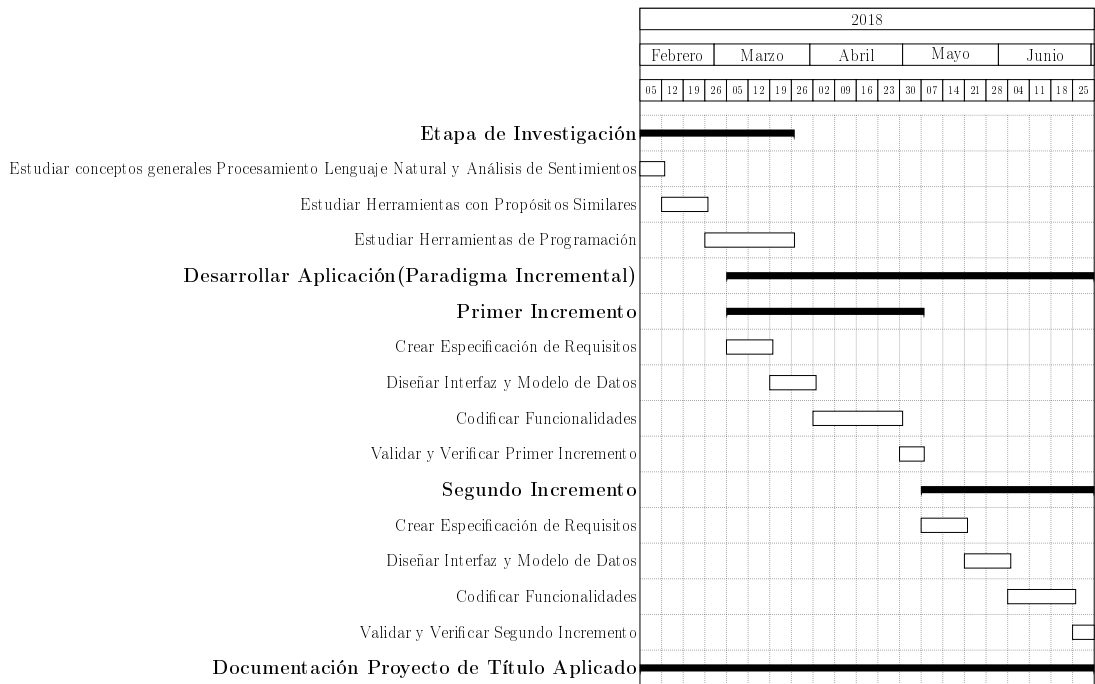
- **Conocimiento HTML:**

Para aumentar la zona objetivo de extracción es necesario que el usuario tenga conocimientos básicos del lenguaje de maquetación HTML el cual es uno de los lenguajes utilizados para la confección de las páginas web.

1.7. Paradigma de Desarrollo de Software

Luego de realizar la etapa de estudio, se continúa con el desarrollo de la aplicación. Se utilizó el Paradigma de Desarrollo Incremental con 2 incrementos, donde el primer incremento implica las funcionalidades con mayor trascendencia para el usuario definidas con anterioridad, en el segundo incremento, se consideraron las funcionalidades restantes para complementar el primer incremento. El modelo de desarrollo incremental permite tener una retroalimentación e interacción del usuario al finalizar cada incremento, permitiendo una comunicación activa entre usuario y desarrollador (Sommerville, 2011).

1.8. Carta Gantt



Capítulo 2

Desarrollo

2.1. Conceptos generales sobre Procesamiento de Lenguaje Natural y Análisis de los Sentimientos.

2.1.1. Procesamiento Lenguaje Natural (PLN).

Una de las tareas fundamentales de la Inteligencia Artificial (IA) es la representación de lenguajes naturales usando herramientas de computación, donde los lenguajes de programación juegan un papel importante, ya que forman el enlace necesario entre los lenguajes naturales y una máquina. El PLN consiste en la utilización de un lenguaje natural para comunicarnos con la computadora, debiendo ésta entender las oraciones que le sean proporcionadas, el uso de estos lenguajes naturales, facilita el desarrollo de programas que realicen tareas relacionadas con el lenguaje o desarrolla modelos que ayudan a comprender los mecanismos humanos relacionados con el lenguaje. El uso del Lenguaje Natural (LN) en la comunicación hombre-máquina presenta a la vez una ventaja y un obstáculo con respecto a otros medios de comunicación (Vásquez et al., 2009), (González, 2004) (Jiménez Zafra et al., 2015)(Martínez et al., 2015).

Ventaja:

La medida que el locutor no tiene que esforzarse para aprender el medio de comunicación a diferencia de otros medios de interacción como lo son los lenguajes de comando o las interfaces gráficas.

Desventaja:

Su uso también presenta limitaciones porque la computadora tiene una limitada comprensión del lenguaje. Por ejemplo, el usuario no puede hablar sobrentendidos, ni introducir nuevas palabras, ni construir sentidos derivados, tareas que se realizan espontáneamente cuando se utiliza el lenguaje natural. Realmente, lo que constituye en ventaja para la comunicación humana se convierte en problema a la hora de un tratamiento computacional, ya que implican conocimiento y procesos de razonamiento que aún no sabemos ni cómo caracterizarlos ni cómo formalizarlos.

Aplicaciones del PLN:

Las aplicaciones del PLN son muy variadas, ya que su alcance es muy grande, algunas de las aplicaciones son:

- Traducción automática.
- Recuperación de la información.
- Extracción de información y resúmenes.
- Comprensión del audio de un reconocimiento de voz.
- Tutores inteligentes.
- Resolución cooperativa de problemas.

2.1.2. Análisis de los Sentimientos (AS).

El Análisis de los Sentimientos es una arista del Procesamiento del Lenguaje Natural que aplica y crea técnicas computacionales para la obtención de información relevante de textos que han sido generados a través de conversiones de los Lenguajes Naturales usados por los Humanos (escrito, hablados, señales) relacionado con el tópico de los sentimientos, el análisis se puede dividir en dos enfoques: Enfoque Afectivo y Enfoque de Polaridad, que buscan caracterizar el contenido sentimental de los textos bajo distintos puntos de vista (Jiménez Zafra et al., 2015) (Martínez et al., 2015).

El análisis de los sentimientos se está aplicando en casi todos los ámbitos comerciales y sociales porque las opiniones son fundamentales en casi todas las actividades humanas y son factores clave que influyen en nuestros comportamientos, tomando un rol importante al momento de una decisión, tanto para individuos como también para organizaciones (Liu, 2012).

Enfoque Afectivo:

El Enfoque Afectivo busca caracterizar el contenido sentimental de los textos a través de teorías psicológicas, como por ejemplo la de Ekman (1993) o la de Plutchik (1980), que postulan emociones que engloban las expresadas por los seres humanos, para finalmente encontrar cuál es la emoción predominante en el contenido del texto.

La teoría psicológica de Ekman (1993), plantea clasificar el aspecto emotivo en 6 emociones básicas tales como miedo, tristeza, alegría, ira, sorpresa y disgusto, la que perfectamente puede ser usada para caracterizar el contenido sentimental de los textos para encontrar cuál de estas emociones es la predominante para posteriormente hacer un juicio sobre el texto.

Enfoque de Polaridad:

El Enfoque de Polaridad analiza el texto proporcionado para determinar si expresa un sentimiento positivo, neutro, negativo o ambiguo (si es imposible detectar algún sentimiento). Para ello, se identifica la polaridad local de las diferentes frases en el texto y se evalúa la relación entre ellas, lo que resulta en un valor de polaridad global para el texto en su conjunto.

Un ejemplo concreto de un Análisis de los Sentimientos es el que realizó el Grupo de Investigadores SOMOS (SOftware MOdelling Science) aplicando el Enfoque Afectivo (Martínez et al., 2015).

2.1.3. Glosario de Términos Relevantes.

Cuando se aborda la necesidad de rescatar información desde la web, se confunden los términos Web Crawler y Web Scraper (o Web Scraping), es por esta razón que se hace alusión a la definición de estos términos para tener una mejor comprensión de cuál puede estar asociado a este proyecto aplicado.

Web Crawler (Rastreador):

Web Crawler son programas diseñados para explorar páginas web en forma automática (TechopediaDefinition-WebCrawler, 2017). La operación normal es que se le da al programa un grupo de direcciones iniciales, el Crawler descarga estas direcciones, analiza las páginas y busca enlaces a páginas nuevas. Luego descarga estas páginas nuevas, analiza sus enlaces, y así sucesivamente, hasta lograr un conjunto de enlaces que tienen algún tipo de relación con las direcciones iniciales, el cual puede ser utilizado para a través de un mecanismo se seleccione el enlace que tiene más aciertos con características especiales de búsqueda otorgadas al Crawler al inicio del proceso, un ejemplo concreto y conocido es Google el Crawler más grande del mundo (GoogleSearch, 2017).

Web Scraper / Web Scraping (Extractor):

Web Scraping es el proceso de recopilar información de forma automática desde la web (TechopediaDefinition-WebScraping, 2017). Usualmente, estos programas simulan la navegación de un humano en la World Wide Web ya sea utilizando el protocolo HTTP manualmente, o incrustando un navegador en una aplicación, se enfoca más en la transformación de datos sin estructura en la web (como el formato HTML) en datos estructurados que pueden ser almacenados y analizados en una base de datos central, en una hoja de cálculo o en alguna otra fuente de almacenamiento.

Algunos de los usos del Web Scraping son:

- Comparación de precios.
- Monitorización de datos climáticos.
- Integración de datos en sitios web.

2.2. Herramientas con propósitos similares (Estado del Arte).

2.2.1. Import.io

Es una aplicación web extractora de datos sin la necesidad de saber programar, la cual genera una fácil interacción con el usuario (Import.io, 2017).

Su funcionamiento consiste de extraer automáticamente información desde alguna página web especificada por el usuario, agrupándolas en celdas, dándole la opción de manipularla, en el caso de que la información otorgada por la herramienta no sea considerada relevante, esta posee una funcionalidad que le permite al usuario crear celdas con información que este va obteniendo desde la visualización de la página mediante clics como se puede apreciar en la Figura 2.1, para luego ser almacenada en algún formato disponible (Excel, CSV, JSON).

Funcionalidades :

- Extracción automática:

Otorga información desde una página con solo ingresar su dirección Url.

- Extracciones programadas:

Ejecuta un extractor en un fecha o tiempo asignado por el usuario.

- Monitoreo de páginas:

Comparación de información extraída de páginas, informando al usuario en el caso de que los datos extraídos sean distintos a extracciones anteriores.

- API para Integración a Apps:

Guarda y crea puntos de acceso en su servidor para otorgar información extraída hacia aplicaciones.

- Descarga de los Datos:

Almacena la información extraída y da opción a descargarla en diferentes formatos (Excel, CSV, JSON).

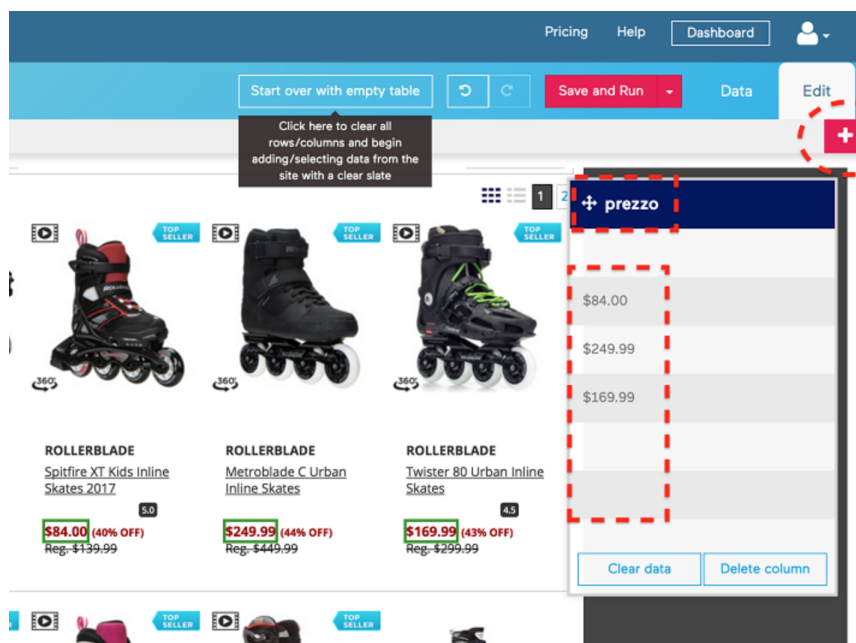


Figura 2.1: Ejemplo: Herramienta Import.io

Posee una licencia pagada, donde dependiendo del paquete comprado se podrán tener acceso a diferentes funcionalidades, también posee una versión gratuita de prueba, pero sus funcionalidades son limitadas (Precios-Import.io, 2017).

2.2.2. ParseHub

Es una aplicación web y una versión de escritorio extractora de datos con interfaz gráfica sin la necesidad de programar (ParseHub, 2017).

Su funcionamiento consiste de crear extractores a partir de una página web, mediante la interacción del usuario con una interfaz donde se muestra la página a extraer, la aplicación genera un diagrama de flujo de acciones realizadas por el usuario, también otorga una visualización de los datos que se van extrayendo, dándole al usuario una panorámica completa de sus acciones en la aplicación, como se puede apreciar en la Figura 2.2.

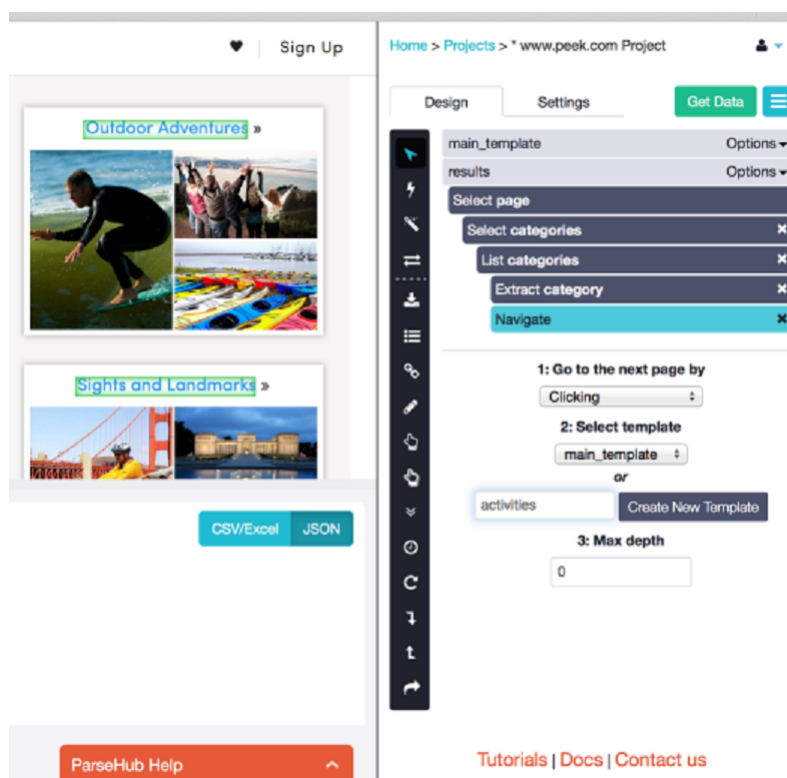


Figura 2.2: Ejemplo: Herramienta ParseHub

Funcionalidades :

- Cloud Storage:

Permite guardar la información extraída en sus servidores alojados en la nube.
- Rotación de IP:

Cambia constantemente de dirección IP para no ser detectado por los Anti-Crawler o Anti-Scraping.
- Extracciones programadas:

Ejecuta un extractor en un fecha o tiempo asignado por el usuario.
- Uso Expresiones Regulares:

Otorga la opción de filtrar la información mediante el uso de expresiones regulares.
- API para Integración a Apps:

Guarda y crea puntos de acceso en su servidor para otorgar información extraída hacia aplicaciones.
- Descarga de los Datos:

Almacena la información extraída y da opción a descargarla en diferentes formatos (Excel, CSV, JSON).

Posee una licencia pagada, donde dependiendo del paquete comprado se podrán tener acceso a diferentes funcionalidades, también posee una versión gratuita, pero se tendrán funcionalidades limitadas (Precios-ParseHub, 2017).

2.2.3. Octoparse

Es una aplicación web extractora de datos sin la necesidad de codificar (Octoparse, 2017).

Su funcionamiento consiste de crear extractores a partir de una página web, mediante la interacción del usuario con una interfaz donde se muestra la página a extraer, la aplicación genera un diagrama de flujo de acciones realizadas por el usuario, también otorga una visualización de los datos que se van extrayendo, dándole al usuario una panorámica completa de sus acciones en la aplicación, como se puede apreciar en la Figura 2.3.

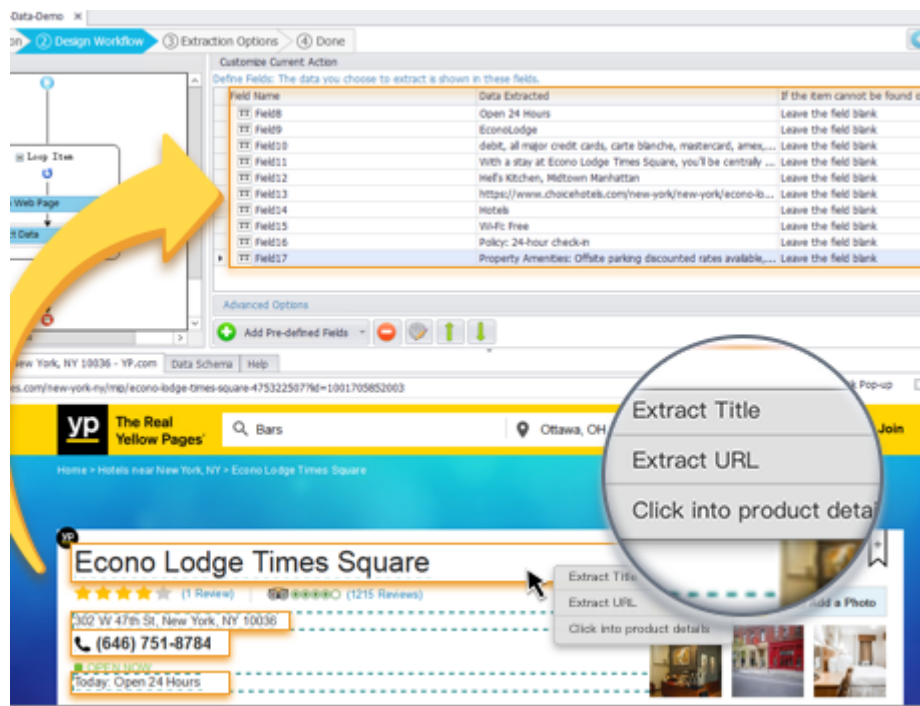


Figura 2.3: Ejemplo: Herramienta Octoparse

Funcionalidades :

- Cloud Storage:

Permite guardar la información extraída en sus servidores alojados en la nube.

- Rotación de IP:

Cambia constantemente de dirección IP para no ser detectado por los Anti-Crawler o Anti-Scraping.

- Extracciones programadas:

Ejecuta un extractor en un fecha o tiempo asignado por el usuario.

- Descarga de los Datos:

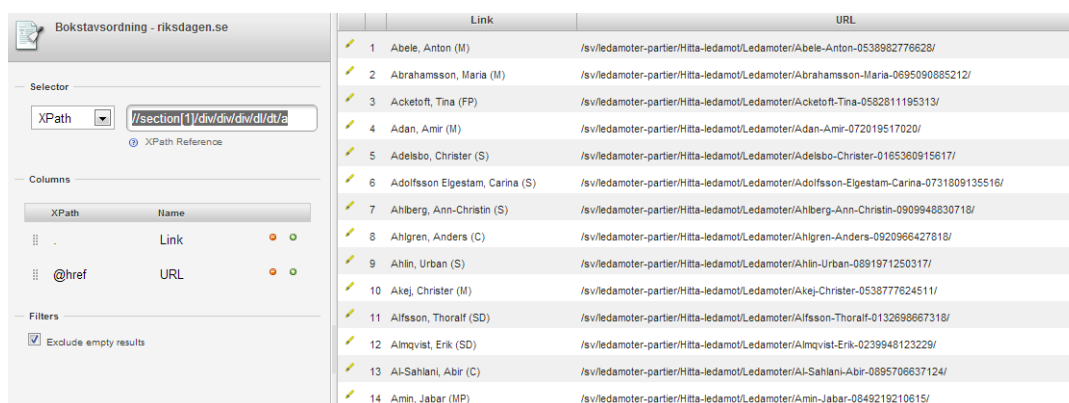
Almacena la información extraída y da opción a descargarla en diferentes formatos (Excel, CSV, JSON).

Posee una licencia pagada, donde dependiendo del paquete comprado se podrán tener acceso a diferentes funcionalidades, también posee una versión gratuita, pero se tendrán funcionalidades limitadas (Precios-Octoparse, 2017).

2.2.4. Scraper Google Extension

Es una extensión para Google Chrome extractora de datos simple y fácil de usar para los que están familiarizados con el lenguaje XPath (ScraperGoogleExtension, 2017).

Su funcionamiento consiste de crear extractores mediante el uso del lenguaje XPath, con el cual le da referencia a la aplicación para extraer los elementos del DOM de la página web desde donde se quiere extraer información, también otorga una visualización de cuáles son los elementos extraídos al aplicar el extractor, como se puede apreciar en la Figura 2.4.



	Link	URL
1	Abele, Anton (M)	/sv/ledamoter-partier/Hitta-ledamoti/Ledamoteri/Abele-Anton-0538982776628/
2	Abrahamsson, Maria (M)	/sv/ledamoter-partier/Hitta-ledamoti/Ledamoteri/Abrahamsson-Maria-0695090885212/
3	Acketoft, Tina (FP)	/sv/ledamoter-partier/Hitta-ledamoti/Ledamoteri/Acketoft-Tina-0562811195313/
4	Adan, Amir (M)	/sv/ledamoter-partier/Hitta-ledamoti/Ledamoteri/Adan-Amir-072019517020/
5	Adelsbo, Christer (S)	/sv/ledamoter-partier/Hitta-ledamoti/Ledamoteri/Adelsbo-Christer-0165360915817/
6	Adolfsson Elgestam, Carina (S)	/sv/ledamoter-partier/Hitta-ledamoti/Ledamoteri/Adolfsson-Elgestam-Carina-0731809135516/
7	Ahlberg, Ann-Christin (S)	/sv/ledamoter-partier/Hitta-ledamoti/Ledamoteri/Ahlberg-Ann-Christin-090994830718/
8	Ahlgren, Anders (C)	/sv/ledamoter-partier/Hitta-ledamoti/Ledamoteri/Ahlgren-Anders-0920966427818/
9	Ahlin, Urban (S)	/sv/ledamoter-partier/Hitta-ledamoti/Ledamoteri/Ahlin-Urban-0891971250317/
10	Akej, Christer (M)	/sv/ledamoter-partier/Hitta-ledamoti/Ledamoteri/Akej-Christer-0538777824511/
11	Alfsson, Thoralf (SD)	/sv/ledamoter-partier/Hitta-ledamoti/Ledamoteri/Alfsson-Thoralf-0132698667318/
12	Almqvist, Erik (SD)	/sv/ledamoter-partier/Hitta-ledamoti/Ledamoteri/Almqvist-Erik-0239948123229/
13	Al-Sahlani, Abir (C)	/sv/ledamoter-partier/Hitta-ledamoti/Ledamoteri/Al-Sahlani-Abir-0895706637124/
14	Amin, Jabar (MP)	/sv/ledamoter-partier/Hitta-ledamoti/Ledamoteri/Amin-Jabar-0849219210615/

Figura 2.4: Ejemplo: Herramienta Scraper Google Extension

Funcionalidades :

- Descarga de los Datos:

Almacena la información extraída y da opción a descargarla en diferentes formatos (Excel, CSV, JSON).

- Extracción de Datos:

Permite la extracción de información desde una página web.

- Uso Lenguaje XPath:

Permite el uso del lenguaje XPath para indicar cuáles son los elementos que se desean extraer mediante una petición al DOM de la página.

Posee una licencia totalmente gratuita y se puede encontrar en la Web Store de las extensiones de Google Chrome (ChromeWebStore, 2017).

2.3. Herramientas de programación

2.3.1. React.js.

React es una librería de Javascript para construir interfaces de usuario interactivas basadas en componentes (ReactJS, 2017), (ReactJS-Articles, 2017).

Sus principales características:

- **Única Página Web:**

Es utilizado para crear páginas web de una única página (Single Page Application) con el propósito de dar una experiencia más fluida a los usuarios como una aplicación de escritorio, cargando recursos dinámicamente como respuesta a las acciones del usuario. La página no tiene que cargar otra vez en ningún punto del proceso.

- **Lenguaje Declarativo:**

Al usar Lenguaje Declarativo para la creación de vistas de cada estado de la página web, hace que el código sea predecible y fácil de depurar. React posee la capacidad de actualizar y crear eficientemente el componente correcto cuando los datos que maneja la página web cambian.

- **Basado en Componentes:**

Permite la creación de componentes encapsulados que pueden manejar su propio estado, siendo útil para la creación de vistas más complejas que necesitan ser compuestas por varios componentes.

La lógica de los componentes se escribe en Javascript junto a JSX sintaxis que acepta crear componentes con características de HTML y Javascript, logrando una fácil manera de traspasar datos importantes a través de toda

la página web. De esta manera se utiliza JSX por sobre los modelos pre-determinados que se pueden crear en HTML (templates) como se puede apreciar en la Figura 2.5.

Primero se crea el componente (HelloMessage), todo lo que se creará cuando el componente sea utilizado debe ir en la función render(), teniendo la salvedad que siempre se debe retornar un solo elemento, si se desea retornar varios elementos se deben agrupar en un elemento mayor, se observa que el elemento contiene un `this.props.name`, 'this' hace referencia al componente en este caso (HelloMessage), 'props' hace referencia a las propiedades del componente, 'name' hace referencia a la propiedad que se desea utilizar, por lo tanto el componente tiene una dependencia de una propiedad traspasada al momento de ser utilizado, convirtiéndose en un componente dinámico dado que si se cambia la propiedad traspasada, este componente automáticamente se actualizará utilizando la nueva propiedad, en la parte inferior de la Figura 2.5 se muestra la sintaxis JSX mencionada anteriormente, se puede apreciar que el componente utilizado es `<HelloMessage>` creado en la parte superior de la Figura 2.5, a este componente se le ha atribuido la propiedad 'name' con el texto 'John', por lo tanto la referencia anterior `this.props.name` tendrá el contenido de un texto 'John', si esto lo compilamos nos mostrará un texto 'Hello John', ¿Qué pasaría si cambiáramos 'John' por 'World'?, nos mostraría un texto 'Hello World'.

```
class HelloMessage extends React.Component {
  render() {
    return (
      <div>
        Hello {this.props.name}
      </div>
    );
  }
}

ReactDOM.render(
  <HelloMessage name="John" />,|
);
```

Figura 2.5: React Component.

2.3.2. Node.js.

Node es una librería de código abierto construida en base al Javascript de tiempo de ejecución utilizada por Chrome para aplicaciones rápidas y escalables en la parte del servidor (NodeJS, 2017), (NodeCollection, 2017).

Sus principales características:

- **Lenguaje Javascript:**

Debido a que Node esta basado en el lenguaje Javascript utilizado por Chrome le permite usar este lenguaje en el lado del servidor dando la posibilidad de realizar proyectos a partir de un solo lenguaje, cubriendo todos los puntos importantes de una aplicación, la base de datos, servidor y la interfaz de usuario.

- **Node Package Manager (NPM):**

La NPM es un servicio que ofrece Node creando un repositorio de intercambio de módulos, dado que es de código abierto, se ha creado una comunidad bastante popular y activa en constante crecimiento, tanto que ya se han creado más de 60.000 módulos para el libre uso.

- **Concurrencia y No-Bloqueo:**

Node trabaja con un único hilo de ejecución basado en la programación orientada a eventos, donde tiene una estructura que le permite realizar acciones asíncronas, pero a la vez tiene la capacidad de que si alguna de estas acciones falla no afecte al proceso principal del ciclo que está esperando a que sucedan eventos, permitiendo una concurrencia de múltiples peticiones al servidor como se puede apreciar en la Figura 2.6, donde se refleja el flujo que hacen las peticiones realizadas al servidor, pasando por el Event Loop, ejecutando la acción requerida por la petición en el servidor, y cuando se obtiene la respuesta se ejecuta una función Callback (función que se ejecuta cuando se recibe alguna respuesta) para traspasar los resultados como respuesta a la petición.

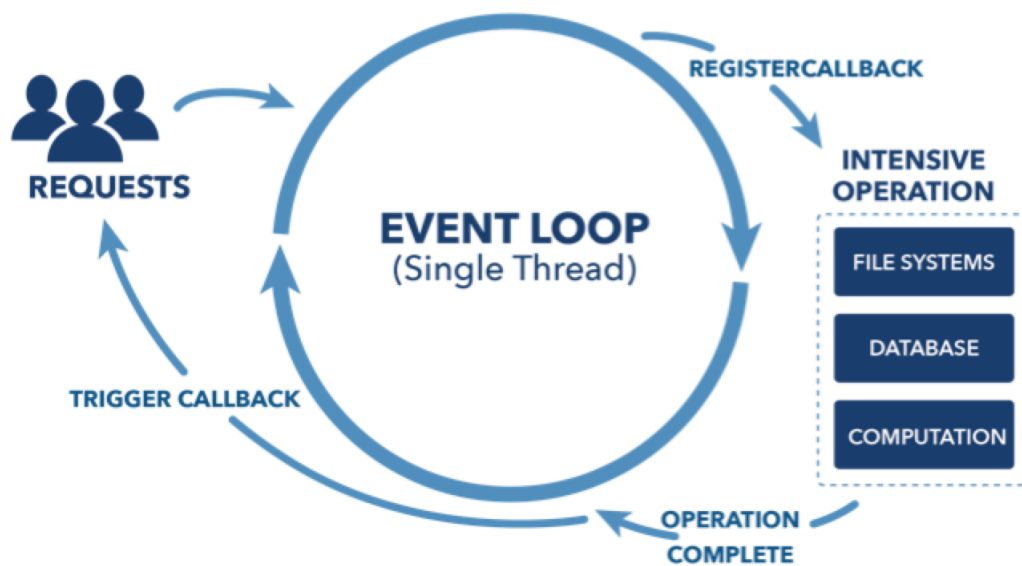


Figura 2.6: Diagrama de Programación Orientada a Eventos.

Node es una de las librerías de las nuevas generaciones que permiten afrontar los problemas desde un punto de vista diferente y con un lenguaje que es conocido e intuitivo como Javascript, captando la atención de muchos desarrolladores.

2.3.3. MongoDB.

MongoDB es una base de datos libre de esquemas, orientada a documentos, escrita en C++. La base de datos está basada en el almacén de documentos, lo que significa que almacena valores (denominados documentos) en forma de datos codificados (MongoDB, 2017), (MongoDBArticles, 2017).

Sus principales características:

- **Rendimiento y Escalabilidad:**

MongoDB tiene un sistema flexible de almacenamiento de esquemas, lo que significa que los objetos almacenados no tienen que tener la misma estructura o los mismos campos, también tiene algunas características de optimización, que distribuye las colecciones de datos, mejorando el rendimiento y consiguiendo un sistema más equilibrado, como no está sujeto a esquemas, si la base de datos llegara a cambiar o crecer, la transición es más sencilla ya que no necesita tener cuidado con las relaciones por su característica de ser una base de datos NO-SQL, ganando escalabilidad pero perdiendo consistencia, la cual estará en manos del programador.

- **Lenguaje de Consultas Javascript:**

Las consultas para leer o escribir en la base de datos de MongoDB se ejecutan en una consola con entorno de Javascript por lo tanto permite usar este lenguaje, como se puede apreciar en la Figura 2.7, donde se aplica la función `find()` a una colección de `USERS` de la base de datos en MongoDB, en busca de todos los usuarios que tengan una edad de 18 años, para luego entregar el nombre de los usuarios como respuesta.

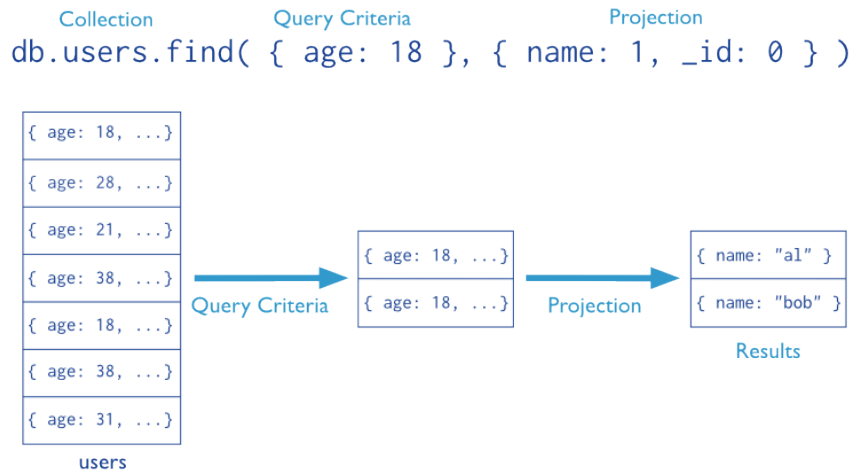


Figura 2.7: Ejemplo: Consulta `find()` MongoDB.

2.3.4. GitHub.

Es una plataforma web construida para desarrolladores, donde se puede almacenar y revisar códigos, manejar proyectos y construir software en conjunto con otros desarrolladores (GitHub, 2017).

Esta plataforma conformada por una gran comunidad, donde se encuentra una gran cantidad de ejemplos codificados de aplicaciones realizadas en distintas herramientas de programación, es utilizada como fuente de estudio e información para las herramientas de programación que fueron utilizadas a lo largo de este proyecto.

2.4. Desarrollo de la aplicación Text Extraction Community

El desarrollo de la aplicación Text Extraction Community se planificó bajo el Paradigma Incremental aplicando 2 incrementos. La aplicación ofrece 3 servicios, los cuales son:

- **Servicio de Extractor desde Twitter.**
- **Servicio de Crear Extractor Personalizado.**
- **Servicio Mis Extractores.**

El Servicio de Crear un Extractor Personalizado se consideró como el más complejo, debido a que involucra varios subprocesos, por esta razón que se eligió para ser desarrollado durante el primer incremento.

El Servicio Mis Extractores se consideró con una complejidad media, ya que reutiliza los procesos del Servicio Crear Extractor Personalizado, con diferencia de que se agregó un par de funcionalidades, es por esta razón que se eligió para ser desarrollado durante el segundo incremento.

El Servicio de Extractor desde Twitter se consideró con complejidad baja, ya que no se necesita una alta interacción con el usuario, lo cual se refleja en su fácil uso, es por esta razón que se eligió para ser desarrollado durante el segundo incremento.

2.4.1. Primer Incremento

Para el primer incremento se contempló el desarrollo de todo lo relacionado con el Servicio de Crear un Extractor Personalizado.

Servicio Crear Extractor Personalizado.

El Servicio Crear Extractor Personalizado ofrece al usuario la posibilidad de caracterizar, especificar, crear y guardar un extractor capaz de capturar textos desde una página web en particular, con la posibilidad de probar, ya sea nodos o el extractor completo, durante el proceso de creación otorgando las opciones de ver y descargar los resultados obtenidos de cada extracción, clasificando los resultados de acuerdo a las características de los nodos o del extractor.

Para aumentar la posibilidad de éxito en el desempeño del extractor o nodos, se recomienda al usuario estudiar la estructura de la página web a la cual desea extraer los textos, debido a que las características necesarias para la creación del extractor, indican donde obtener los textos y cuáles son los que debe extraer.

Terminología.

Para el desarrollo del primer y segundo incremento se consideró la importancia de los siguientes conceptos:

- **Modelo de Objetos del Documento (DOM, siglas en inglés):**

Es la estructura que define los objetos, elementos y la manera en que estos se relacionan para dar como resultado una página web, estos objetos y elementos pueden estar caracterizados con propiedades que hacen posible referenciarlos y manipularlos, el lenguaje de etiquetas generalmente utilizado es HTML, se puede ingresar al DOM de cualquier página web apretando click derecho seleccionando la opción Ver código fuente de la página o Inspeccionar elemento.

- **Estructura del Extractor:**

El Extractor esta conformado por Nodos Extractores, que a su vez están compuestos por un Contenedor y Elementos Extractores, cada Elemento posee una Etiqueta que deriva una Ruta.

- **Nodo Extractor:**

Objeto compuesto por un Contenedor y Elementos Extractores que permiten seleccionar y extraer textos desde elementos de una sección específica de la página web.

- **Contenedor:**

Objeto estructurado compuesto por elementos potencialmente extraíbles, por lo general se repite en páginas correctamente indexadas, facilitando su identificación, es el Objeto o Elemento que contiene los textos deseados para la extracción.

- **Elemento Extractor:**

Objeto compuesto por una Etiqueta y Ruta, permitiendo identificar los elementos potencialmente a extraer desde los Contenedores.

- **Etiqueta:**

Nombre de rotulación para identificar una Ruta, sirve para clasificar los datos cuando se visualiza o descarga los resultados de extracción.

- **Ruta:**

Característica o propiedad que permite referenciar o manipular un elemento específico desde el Contenedor.

2.4.1.1. Especificación de Requisitos

2.4.1.1.1 Requisitos Funcionales

Los requisitos funcionales son propiedades o características que establecen el comportamiento del sistema, se representan a través de funciones descritas con un nombre, una breve descripción, conjunto de entradas, proceso o comportamiento y conjunto de salidas, además tienen como fin ayudar al lector durante el desarrollo o evaluación del sistema.

A continuación se presentan los requisitos funcionales desarrollados durante el primer incremento de la aplicación:

- **R1. Crear Elemento Extractor**
- **R2. Eliminar Elemento Extractor**
- **R3. Crear Nodo Extractor**
- **R4. Probar Nodo Extractor**
- **R5. Editar Nodo Extractor**
- **R6. Guardar Nodo Extractor**
- **R7. Eliminar Nodo Extractor**
- **R8. Crear Extractor Personalizado**
- **R9. Guardar Extractor Personalizado**
- **R10. Probar Extractor Personalizado**
- **R11. Descargar Datos Extraídos por Nodos**
- **R12. Visualizar Datos Extraídos por Nodos**
- **R13. Descargar Datos Extraídos por Extractores**
- **R14. Visualizar Datos Extraídos por Extractores**

R1. Crear Elemento Extractor

■ Descripción:

Permite crear un elemento capaz de dar información al nodo extractor para encontrar los elementos a extraer desde el DOM de una página web, mediante el identificador **etiqueta** y el enlace al DOM **ruta**.

■ Entradas y Restricciones:

Etiqueta (String 4-30 caracteres obligatorio).

Ruta (String 1-255 caracteres obligatorio).

Elementos Extractores (Object 1-500 keys-values obligatorio).

■ Proceso:

R1.1: Se ingresa la etiqueta y ruta para luego iniciar la creación del elemento extractor a través del botón Crear.

R1.2: Si la etiqueta ingresada es única, se crea el elemento extractor y se agrega al objeto de elementos extractores, de lo contrario se muestra un mensaje de advertencia: 'Error al crear elemento extractor, la etiqueta ya existe'.

■ Salida:

Creación elemento extractor exitosa o mensaje de advertencia.

R2. Eliminar Elemento Extractor

- **Descripción:**

Permite eliminar un elemento desde los elementos extractores, mediante el identificador **etiqueta**.

- **Entradas y Restricciones:**

Etiqueta (String 4-30 caracteres obligatorio).

Elementos Extractores (Object 1-500 keys-values obligatorio).

- **Proceso:**

R2.1: Se selecciona el elemento extractor desde la visualización de los elementos extractores a través del botón X presente en cada elemento extractor, luego se eliminará toda la información relacionada al elemento extractor seleccionado en el objeto de los elementos extractores, por lo tanto desaparece de la visualización de los elementos extractores.

- **Salida:**

Eliminación de elemento extractor exitosa.

R3. Crear Nodo Extractor

- **Descripción:**

Permite crear un nodo capaz de extraer elementos de una sección del DOM de una página web, mediante un **contenedor** y **elementos extractores**.

- **Entradas y Restricciones:**

Nombre del Nodo (String 4-50 caracteres obligatorio).

Descripción (String 4-255 caracteres obligatorio).

Contenedor (String 4-255 caracteres obligatorio).

Elementos Extractores (Object 1-500 keys-values obligatorio).

Nodos Extractores (Object 1-500 keys-values obligatorio).

- **Proceso:**

R3.1: Se ingresa el contenedor y los elementos extractores, luego se ingresa una descripción y el nombre del nodo para posteriormente iniciar la creación del nodo extractor a través del botón Guardar.

R3.2: Si el nombre del nodo ingresado es único, se crea el nodo extractor y se agrega al objeto de los nodos extractores mostrando el mensaje de éxito: 'El nodo ha sido guardado con éxito!.', de lo contrario se muestra un mensaje de advertencia: 'Error al guardar el nodo extractor, el nombre del nodo ya existe.'.

- **Salida:**

Creación de nodo extractor exitosa o mensaje de advertencia.

R4. Probar Nodo Extractor

- **Descripción:**

Permite comunicarse con el Servicio de Extracción Personalizada - Probar Nodo, para comprobar si las características del nodo realmente son correctas y capaces de extraer el texto que se desea desde una página web, la función puede utilizarse en los procesos de Crear y Editar un nodo extractor.

- **Entradas y Restricciones:**

Url (String 16-255 caracteres obligatorio).

Nombre del Nodo (String 4-50 caracteres obligatorio).

Descripción (String 4-255 caracteres obligatorio).

Contenedor (String 4-255 caracteres obligatorio).

Elementos Extractores (Object 1-500 keys-values obligatorio).

- **Proceso:**

R4.1: Se ingresa la Url. nombre del nodo, descripción, contenedor y los elementos extractores para luego iniciar el proceso de probar el nodo extractor a través del botón Probar Nodo, el cual ingresa los datos al Servicio de Probar Nodo presente en el servidor.

R4.2: Si la prueba resulta exitosa, aparecerán botones para visualizar o descargar los resultados extraídos, de lo contrario se muestra un mensaje de advertencia: 'Ha ocurrido un error al probar el nodo extractor. no se han encontrado resultados, revisa los parámetros, luego, intenta nuevamente.'

R4.3: En el caso de error de conexión con el servicio, se muestra el mensaje de advertencia: 'Ha ocurrido un error al probar el nodo extractor, no se ha podido conectar con el Servicio - Probar Nodo.'

- **Salida:**

Resultados extraídos o mensaje de advertencia.

R5. Editar Nodo Extractor

- **Descripción:**

Permite modificar las características de un nodo ya existente.

- **Entradas y Restricciones:**

Nombre del Nodo (String 4-50 caracteres obligatorio).

Descripción (String 4-255 caracteres obligatorio).

Contenedor (String 4-255 caracteres obligatorio).

Elementos Extractores (Object 1-500 keys-values obligatorio).

Nodos Extractores (Object 1-500 keys-values obligatorio).

- **Proceso:**

R5.1: Se selecciona un nodo extractor desde la visualización de los nodos extractores a través de clicar sobre un nodo, posteriormente se abrirá una visualización con todas las características del nodo seleccionado dando la oportunidad de modificarlas, luego de modificar, se inicia la modificación de las características del nodo a través del botón Guardar Nodo.

R5.2: Si el nombre del nodo modificado es único, se modifican las características en el nodo seleccionado a partir de la visualización de los nodos extractores mostrando el mensaje de éxito: 'El nodo ha sido guardado con éxito!.', de lo contrario se muestra el mensaje de advertencia: 'Error al guardar el nodo extractor, el nombre del nodo ya existe.'

R5.3: En el caso de guardar las modificaciones pero estas no poseen elementos extractores, se muestra el mensaje de advertencia: 'El nodo debe tener al menos 1 elemento extractor.'

- **Salida:**

Modificaciones del nodo extractor exitosa o mensaje de advertencia.

R6. Guardar Nodo Extractor

- **Descripción:**

Permite almacenar las características de un nodo extractor en los nodos extractores.

- **Entradas y Restricciones:**

Nombre del Nodo (String 4-50 caracteres obligatorio).

Descripción (String 4-255 caracteres obligatorio).

Contenedor (String 4-255 caracteres obligatorio).

Elementos Extractores (Object 1-500 keys-values obligatorio).

Nodos Extractores (Object 1-500 keys-values obligatorio).

- **Proceso:**

R6.1: Se ingresa el contenedor y los elementos extractores, luego se ingresa una descripción y el nombre del nodo para posteriormente iniciar el guardado de las características del nodo extractor a través del botón Guardar Nodo.

R6.2: Si el nombre del nodo ingresado es único, se guarda el nodo extractor y se agrega al objeto de los nodos extractores mostrando el mensaje de éxito: 'El nodo ha sido guardado con éxito!.', de lo contrario se muestra un mensaje de advertencia: 'Error al guardar el nodo extractor, el nombre del nodo ya existe.'.

R6.3: En el caso de guardar las características del nodo pero estas no poseen elementos extractores, se muestra el mensaje de advertencia: 'El nodo debe tener al menos 1 elemento extractor.'

- **Salida:**

Guardar nodo extractor exitoso o mensaje de advertencia.

R7. Eliminar Nodo Extractor

- **Descripción:**

Permite borrar las características de un nodo extractor existente.

- **Entradas y Restricciones:**

Nombre del Nodo (String 4-50 caracteres obligatorio).

Nodos Extractores (Object 1-500 keys-values obligatorio).

- **Proceso:**

R7.1: Se selecciona un nodo extractor desde la visualización de los nodos extractores a través de clicar sobre un nodo, posteriormente se abrirá una visualización con todas las características del nodo seleccionado, luego se inicia la eliminación a través del botón Eliminar Nodo, el cual abrirá una confirmación de eliminación, en el caso de aceptar la confirmación se procederá a eliminar el nodo seleccionado en el objeto de los nodos extractores mostrando el mensaje de éxito: 'El nodo ha sido eliminado con éxito!.', en el caso de rechazar la confirmación se dará por terminado el proceso.

- **Salida:**

Eliminación de nodo extractor exitosa.

R8. Crear Extractor Personalizado

■ Descripción:

Permite caracterizar un extractor, para que al ejecutarlo o probarlo, sea capaz de extraer el texto de secciones y elementos específicos desde el DOM de una página web.

■ Entradas y Restricciones:

Usuario (String 4-50 caracteres obligatorio).

Url (String 16-255 caracteres obligatorio).

Nombre del Extractor (String 4-50 caracteres obligatorio).

Descripción (String 4-255 caracteres obligatorio).

Nodos Extractores (Object 1-500 keys-values obligatorio).

■ Proceso:

R8.1: Se ingresa el usuario y Url, luego se crean los nodos extractores para posteriormente ingresar una descripción y el nombre del extractor iniciando la creación del extractor a través del botón Guardar Extractor.

R8.2: Si el nombre del extractor ingresado es único, se crea el extractor y se agrega al objeto de los extractores del usuario mostrando el mensaje de éxito: 'El extractor ha sido guardado con éxito!.', de lo contrario se muestra un mensaje de advertencia: 'Error al guardar el extractor, el nombre del extractor ya existe.'

R8.3: En el caso de crear las características del extractor pero estas no poseen nodos extractores, se muestra el mensaje de advertencia: 'El extractor debe tener al menos 1 nodo extractor.'

■ Salida:

Creación del extractor exitosa o mensaje de advertencia.

R9. Guardar Extractor Personalizado

■ Descripción:

Permite almacenar en la base de datos las características de un extractor, asociándolo al usuario que lo creo, otorgando la posibilidad de ser ocupado nuevamente mediante el Servicio Mis Extractores.

■ Entradas y Restricciones:

Usuario (String 4-50 caracteres obligatorio).

Url (String 16-255 caracteres obligatorio).

Nombre del Extractor (String 4-50 caracteres obligatorio).

Descripción (String 4-255 caracteres obligatorio).

Nodos Extractores (Object 1-500 keys-values obligatorio).

■ Proceso:

R9.1: Se ingresa el usuario y Url, luego se crean los nodos extractores para posteriormente ingresar una descripción y el nombre del extractor iniciando el guardado de las características del extractor a través del botón Guardar Extractor.

R9.2: Si el nombre del extractor ingresado es único, se guarda el extractor y se agrega al objeto de los extractores del usuario mostrando el mensaje de éxito: 'El extractor ha sido guardado con éxito!.', de lo contrario se muestra un mensaje de advertencia: 'Error al guardar el extractor, el nombre del extractor ya existe.'

R9.3: En el caso de guardar las características del extractor pero estas no poseen nodos extractores, se muestra el mensaje de advertencia: 'El extractor debe tener al menos 1 nodo extractor.'

■ Salida:

Guardar características del extractor exitosa o mensaje de advertencia.

R10. Probar Extractor Personalizado

- **Descripción:**

Permite comunicarse con el Servicio de Extracción Personalizada - Probar Extractor, para comprobar si las características del extractor realmente son correctas y capaces de extraer el texto que se desea desde una página web, dividiendo los resultados obtenidos a partir de los nodos extractores que lo conforman.

- **Entradas y Restricciones:**

Usuario (String 4-50 caracteres obligatorio).

Url (String 16-255 caracteres obligatorio).

Nombre del Extractor (String 4-50 caracteres obligatorio).

Descripción (String 4-255 caracteres obligatorio).

Nodos Extractores (Object 1-500 keys-values obligatorio).

- **Proceso:**

R10.1: Se ingresa la Url, nombre del extractor, descripción y los nodos extractores para luego iniciar el proceso de probar el extractor a través del botón Probar Extractor, el cual ingresa los datos al Servicio de Probar Extractor presente en el servidor.

R10.2: Si la prueba resulta exitosa, aparecerán botones para visualizar o descargar los resultados extraídos, de lo contrario se muestra un mensaje de advertencia: 'Ha ocurrido un error al probar el extractor. no se han encontrado resultados, revisa los parámetros, luego, intenta nuevamente.'

R10.3: En el caso de error de conexión con el servicio, se muestra el mensaje de advertencia: 'Ha ocurrido un error al probar el extractor, no se ha podido conectar con el Servicio - Probar Extractor.'

- **Salida:**

Resultados extraídos o mensaje de advertencia.

R11. Descargar Datos Extraídos por Nodos

- **Descripción:**

Da la opción al usuario de poder almacenar los datos extraídos al probar un nodo en formato Texto y CSV, dividiendo la información a partir de los elementos extractores que componen al nodo.

- **Entradas y Restricciones:**

Nombre del Nodo (String 4-50 caracteres obligatorio).

Elementos Extractores (Object 1-500 keys-values obligatorio).

Resultados (Array of Objects 1-500 objetos obligatorio).

- **Proceso:**

R11.1: Se ingresa el nombre del nodo, los elementos extractores y los resultados extraídos al probar el nodo, luego se abrirá una visualización donde se da la opción al usuario de seleccionar el formato de descarga de los resultados, puede seleccionar entre Texto o CSV, posteriormente se generará el archivo correspondiente a la opción seleccionada, iniciando la descarga para el usuario.

- **Salida:**

Archivo.txt o Archivo.csv.

R12. Visualizar Datos Extraídos por Nodos

- **Descripción:**

Da la opción al usuario de poder observar cuáles fueron los datos extraídos por el nodo extractor, dividiendo la información a partir de los elementos extractores que componen al nodo.

- **Entradas y Restricciones:**

Nombre del Nodo (String 4-50 caracteres obligatorio).

Elementos Extractores (Object 1-500 keys-values obligatorio).

Resultados (Array of Objects 1-500 objetos obligatorio).

- **Proceso:**

R12.1: Se ingresa el nombre del nodo, los elementos extractores y los resultados extraídos al probar el nodo, luego se abrirá una visualización donde se despliegan todos los resultados extraídos al probar el nodo, clasificando los resultados a partir de los elementos extractores que componen al nodo extractor.

- **Salida:**

Resultados extraídos.

R13. Descargar Datos Extraídos por Extractores

- **Descripción:**

Da la opción al usuario de poder almacenar los datos extraídos al probar un extractor en formato Texto y CSV, dividiendo la información a partir de los nodos extractores que componen al extractor.

- **Entradas y Restricciones:**

Nombre del Extractores (String 4-50 caracteres obligatorio).

Nodos Extractores (Object 1-500 keys-values obligatorio).

Resultados (Object of Objects 1-500 objetos obligatorio).

- **Proceso:**

R13.1: Se ingresa el nombre del extractor, los nodos extractores y los resultados extraídos al probar el extractor, luego se abrirá una visualización donde se da la opción al usuario de seleccionar el formato de descarga de los resultados, puede seleccionar entre Texto o CSV, posteriormente se generará el archivo correspondiente a la opción seleccionada, iniciando la descarga para el usuario.

- **Salida:**

Archivo.txt o Archivo.csv.

R14. Visualizar Datos Extraídos por Extractores

- **Descripción:**

Da la opción al usuario de poder observar cuáles fueron los datos extraídos por el extractor, dividiendo la información a partir de los nodos extractores que componen al extractor.

- **Entradas y Restricciones:**

Nombre del Extractores (String 4-50 caracteres obligatorio).

Nodos Extractores (Object 1-500 keys-values obligatorio).

Resultados (Object of Objects 1-500 objetos obligatorio).

- **Proceso:**

R14.1: Se ingresa el nombre del extractor, los nodos extractores y los resultados extraídos al probar el extractor, luego se abrirá una visualización donde se despliegan todos los resultados extraídos al probar el extractor, clasificando los resultados a partir de los nodos extractores que componen al extractor.

- **Salida:**

Resultados extraídos.

2.4.1.1.2 Requisitos no Funcionales

Los requisitos no funcionales son propiedades o características que afectan el grado de satisfacción del sistema, se pueden representar a través de restricciones, las cuales son impuestas por el cliente como guía o definición para el sistema, y atributos de calidad que son importantes para los stakeholders lo que puede afectar tanto a la arquitectura del sistema como al grado de satisfacción con el cliente.

A continuación se presentan los requisitos no funcionales considerados para la aplicación:

- **Usabilidad:**

- El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 24 horas.
- El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final.
- El sistema debe poseer información sobre cómo utilizar los Servicios de la aplicación.

- **Rendimiento:**

- El tiempo de respuesta de los Servicios Twitter y Personalizado deben ser menor a 150 segundos.
- El tiempo de respuestas y conexiones con la base de datos deben ser menor a 3.5 segundos para no entorpecer la experiencia del usuario con la aplicación, con excepción de los Servicios ya que están basados en simulación de usuarios para extraer la información.

2.4.1.2. Diseño de Interfaz y Modelo de Datos

2.4.1.2.1 Mockups

Un Mockup es una representación visual y estática de un diseño. En el campo del diseño web los Mockups se utilizan para que los miembros de equipo tengan una idea más clara de cómo lucirá el sitio web.

A continuación se presentan los Mockups desarrollados durante el primer incremento de la aplicación:

Se decidió no crear Mockups para cada funcionalidad, por lo tanto se eligieron las funcionalidades más importantes y que aportaban más contenido a la aplicación.

- **I1. Extractor**
- **I2. Crear Nodo Extractor**
- **I3. Guardar Nodo Extractor**
- **I4. Editar Nodo Extractor**
- **I5. Guardar Extractor**

11. Extractor

La interfaz del Extractor representada en la Figura 2.8, es la sección de la aplicación donde se pueden crear componentes y características para extraer textos de una página web en particular.

El primer paso es especificar la página web de donde extraer los textos, debe ser con el formato completo (ej: <http://www...> o <https://www...>). El segundo paso es empezar a crear nodos, luego de crear por lo menos 1 nodo, se podrá editar nodos, guardar o probar el extractor, en el caso de probar el extractor, aparecerán botones al costado izquierdo representando los resultados obtenidos al ejecutar la extracción, dando la opción de visualizar o descargar los resultados obtenidos, los botones hacen referencia tanto al probar el extractor como al probar los nodos individualmente a partir de la opción editar nodo (clickeando el nombre del nodo).

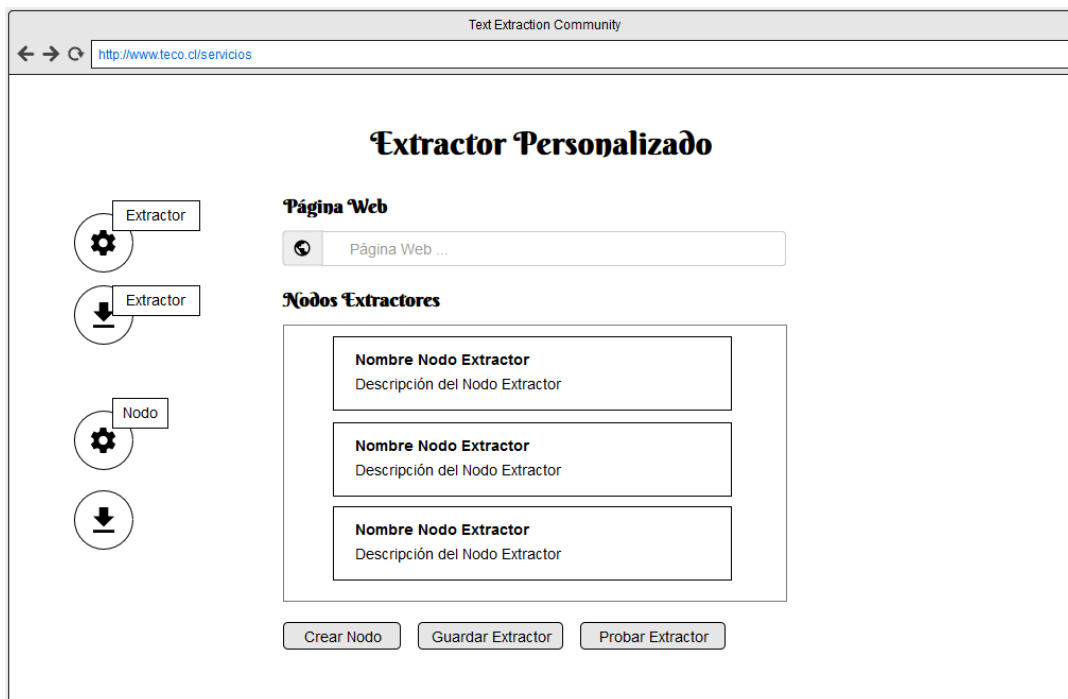
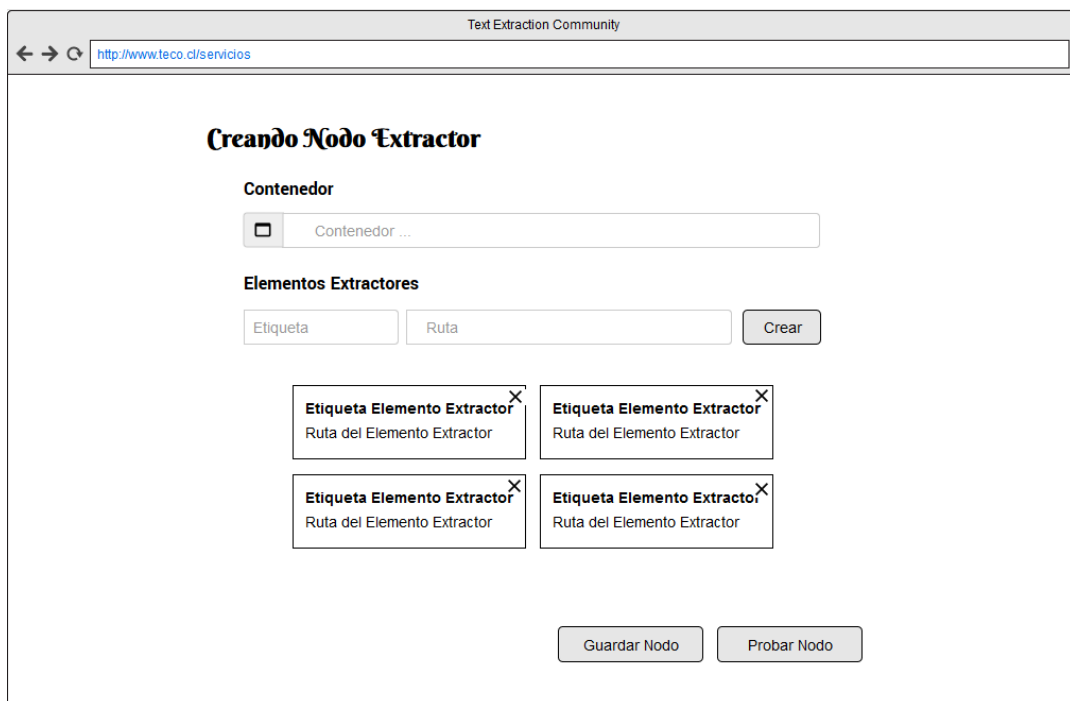


Figura 2.8: Interfaz Extractor.

I2. Crear Nodo Extractor

La interfaz de Crear un Nodo Extractor representada en la Figura 2.9, es la sección de la aplicación donde se pueden crear componentes y características para personalizar un nodo.

El primer paso es especificar el contenedor, posteriormente, crear elementos que puedan indicar cuáles extraer desde el contenedor, luego de crear por lo menos 1 elemento, aparecerán las opciones de eliminar elementos (clickeando el icono X del elemento), guardar o probar el nodo, como se puede apreciar en la Figura 2.9.



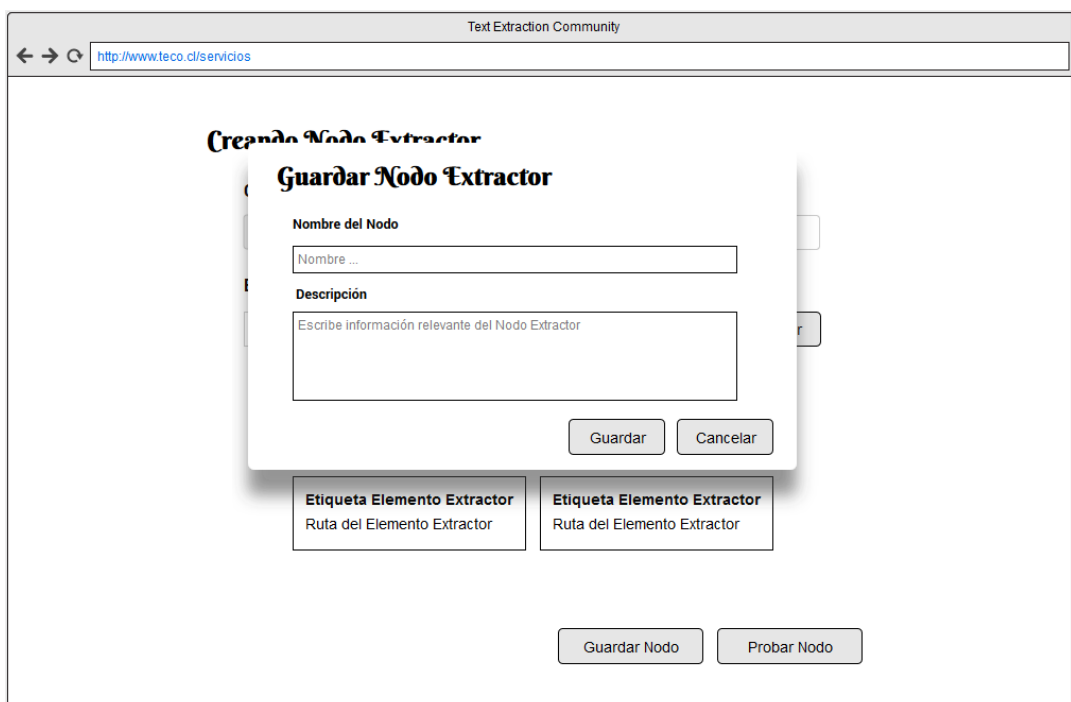
The screenshot shows a web browser window titled "Text Extraction Community" with the URL "http://www.teco.cl/servicios". The main heading is "Creando Nodo Extractor". Under the "Contenedor" section, there is a text input field labeled "Contenedor ...". The "Elementos Extractores" section contains two input fields: "Etiqueta" and "Ruta", followed by a "Crear" button. Below these are four preview cards, each with a close button (X) and the text "Etiqueta Elemento Extractor" and "Ruta del Elemento Extractor". At the bottom, there are two buttons: "Guardar Nodo" and "Probar Nodo".

Figura 2.9: Interfaz Crear Nodo Extractor.

I3. Guardar Nodo Extractor

La interfaz de Guardar un Nodo Extractor representada en la Figura 2.10, es la sección de la aplicación donde se piden las últimas características para guardar y agregar el nodo como parte de un extractor.

El primer paso es especificar el nombre y descripción para el nodo, posteriormente, apretar el botón Guardar, como se puede apreciar en la Figura 2.10.



The screenshot shows a web browser window titled "Text Extraction Community" with the URL "http://www.teco.cl/servicios". The main content area displays a modal dialog box titled "Guardar Nodo Extractor". Inside the dialog, there are two input fields: "Nombre del Nodo" with a placeholder "Nombre ..." and "Descripción" with a placeholder "Escribe información relevante del Nodo Extractor". Below these fields are two buttons: "Guardar" and "Cancelar".

Below the dialog box, there are two identical boxes, each titled "Etiqueta Elemento Extractor" and containing the text "Ruta del Elemento Extractor". At the bottom of the page, there are two buttons: "Guardar Nodo" and "Probar Nodo".

Figura 2.10: Interfaz Guardar Nodo Extractor.

I4. Editar Nodo Extractor

La interfaz de Editar un Nodo Extractor representada en la Figura 2.11, es la sección de la aplicación donde se pueden modificar los componentes y características de un nodo que ya ha sido creado, para ingresar a esta interfaz es necesario clicar un nodo perteneciente a un extractor como se puede apreciar en la Figura 2.8.

La interfaz contiene todas las características del nodo que ha sido clickeado agregando eliminar nodo como una nueva opción , representada en la Figura 2.11.

Text Extraction Community

← → ↻ <http://www.teco.cl/servicios>

Nodo: Nombre del Nodo Clickeado ×

Nombre del Extractor
Nombre del Nodo Clickeado

Descripción
Descripcion del Nodo Clickeado

Contenedor
Contenedor del Nodo Clickeado

Elementos Extractores
Etiqueta Ruta

Etiqueta Elemento Extractor ×
Ruta del Elemento Extractor

Etiqueta Elemento Extractor ×
Ruta del Elemento Extractor

Etiqueta Elemento Extractor ×
Ruta del Elemento Extractor

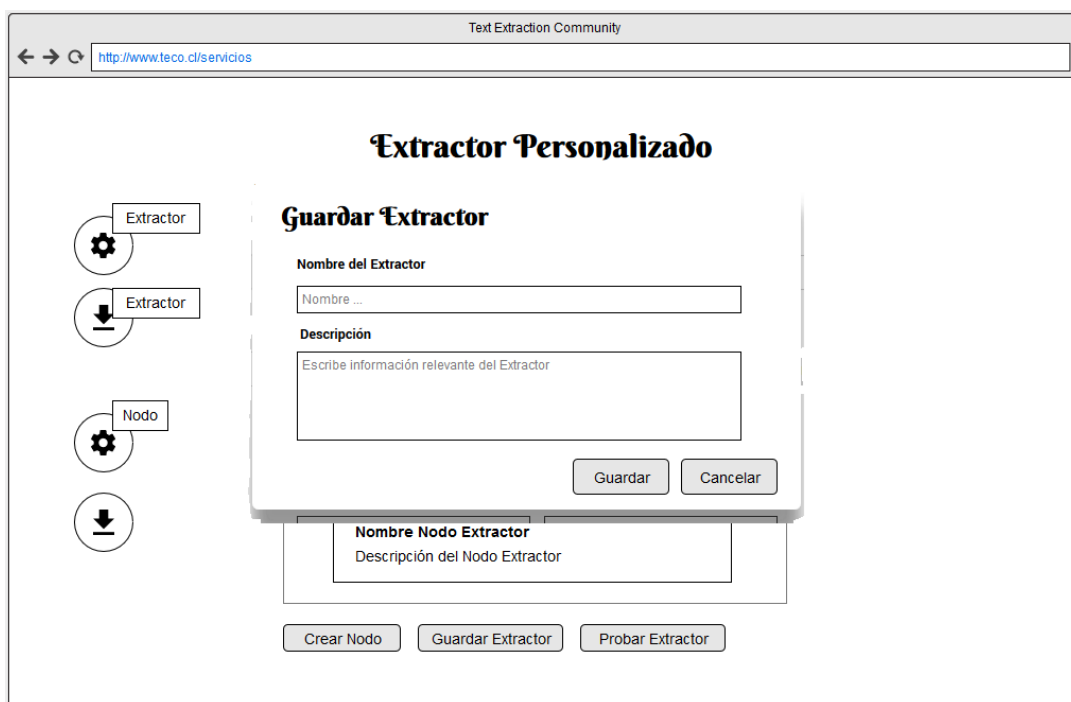
Etiqueta Elemento Extractor ×
Ruta del Elemento Extractor

Figura 2.11: Interfaz Editar Nodo Extractor.

I5. Guardar Extractor

La interfaz de Guardar un Extractor representada en la Figura 2.12, es la sección de la aplicación donde se piden las últimas características para guardar y agregar el extractor a la lista de extractores del usuario en la base de datos.

El primer paso es especificar el nombre y descripción para el Extractor, posteriormente, apretar el botón Guardar, como se puede apreciar en la Figura 2.12.



The screenshot shows a web browser window titled "Text Extraction Community" with the URL "http://www.teco.cl/servicios". The main heading is "Extractor Personalizado". On the left, there is a sidebar with four icons: a gear labeled "Extractor", a downward arrow labeled "Extractor", a gear labeled "Nodo", and a downward arrow. The main content area is titled "Guardar Extractor" and contains a form with two fields: "Nombre del Extractor" (with a placeholder "Nombre ...") and "Descripción" (with a placeholder "Escribe información relevante del Extractor"). Below these fields are "Guardar" and "Cancelar" buttons. At the bottom of the form, there is a section for "Nombre Nodo Extractor" and "Descripción del Nodo Extractor". Below the form, there are three buttons: "Crear Nodo", "Guardar Extractor", and "Probar Extractor".

Figura 2.12: Interfaz Guardar Extractor.

2.4.1.2.2 Diagrama de Actividades

Un diagrama de actividades muestra procesos del software como un flujo de trabajo a través de una serie de acciones que varían dependiendo del grado de interacción con el sistema (Microsoft, 2017).

A continuación se presentan los diagramas de actividades desarrollados durante el primer incremento de la aplicación:

D1. Diagrama de Actividades Extractor Personalizado

El diagrama del Extractor Personalizado muestra el flujo de las actividades necesarios para crear un extractor con éxito, considerando las actividades relacionadas con los nodos y elementos extractores, como también las actividades de visualización y descarga de resultados obtenidos desde extracciones realizadas probando el extractor o nodos, como se puede apreciar en la Figura 2.13.

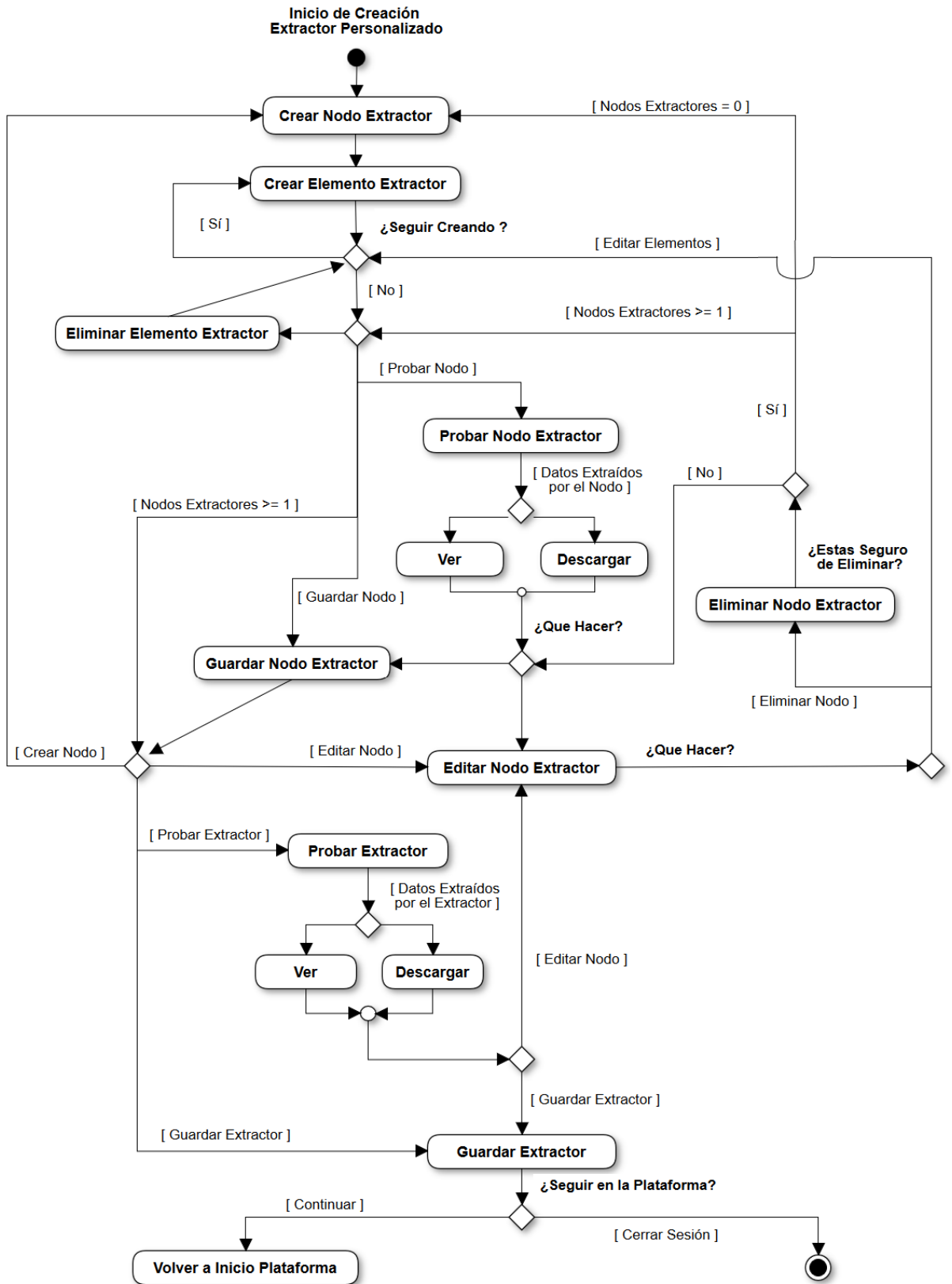


Figura 2.13: Diagrama de Actividades Extractor Personalizado.

2.4.1.3. Codificación

Para el proceso de codificación de las funciones de la aplicación se ocupó el entorno de Node.js (NodeJS, 2017), lo que conlleva el uso constante de la consola o terminal del sistema, es por esta razón que se utilizó el editor de texto Visual Studio Code (VisualStudioCode, 2017), ya que entre sus funciones permite abrir una consola o terminal en la carpeta del proyecto la cual se integra a la interfaz del editor de texto, facilitando la experiencia del usuario.

El uso del entorno de Node permite conexión directa con la Node Package Manager (NPM) con la posibilidad de beneficiarse de las dependencias, módulos o paquetes que existentes en la plataforma.

Para la codificación de las funciones desarrolladas durante el primer incremento se utilizaron las siguientes dependencias:

- **Yarn :**

Disminuye el tiempo de instalación de las dependencias, maximizando recursos (Github-Yarn, 2017).

- **Create React App :**

Crea un entorno configurado de herramientas necesarias para un proyecto en React, dando la posibilidad de concentrarse en el código del proyecto inmediatamente (Github-Create-React-App, 2017).

- **axios (versión 0.16.2):**

Permite realizar peticiones HTTP, transformando la respuesta automáticamente a formato JSON (Github-Axios, 2017).

- **react (versión 15.6.1):**

Permite usar las características de React en el proyecto (Componentes, JSX, etc.) (Github-React, 2017).

- **react-bootstrap (versión 0.31.2):**

Permite ocupar componentes creados por Bootstrap para ser utilizados con React (Github-React-Bootstrap, 2017).

- **semantic-ui-react (versión 0.71.5):**

Permite ocupar componentes creados por Semantic UI para ser utilizados con React (Github-Semantic-UI-React, 2017).

- **redux (versión 3.7.2):**

Permite crear un contenedor State general para aplicaciones en JavaScript (Github-Redux, 2017).

- **redux-thunk (versión 2.2.0):**

Permite enviar peticiones asincrónicas desde las Acciones de Redux (Github-Redux-Thunk, 2017).

- **redux-logger (versión 3.0.6):**

Herramienta para mostrar el actuar de una acción de Redux en la consola del navegador (Github-Redux-Logger, 2017).

- **react-redux (versión 5.0.6):**

Permite utilizar Bindings para conectar componentes de React con Redux (Github-React-Redux, 2017).

- **react-csv (versión 1.0.8):**

Permite crear un documento CSV de acuerdo a datos traspasados como propiedad del componente (Github-React-CSV, 2017).

- **react-router (versión 4.2.0):**

Permite utilizar Componentes de Navegación para crear formas de navegación en la aplicación creada en React (Github-React-Router, 2017).

- **react-dom (versión 15.6.1):**

Permite crear en el DOM de la aplicación los renders de los componentes de React (Github-React-DOM, 2017).

- **react-router-dom (versión 4.1.2):**

DOM Bindings para React Router (Github-React-Router-DOM, 2017).

En el Primer Incremento se codificó las funciones relacionadas con el Servicio de Crear un Extractor Personalizado.

Se creó un componente para el Extractor Personalizado, donde maneja estados que le permite manipular componentes internos de acuerdo a la interacción del usuario, como se puede apreciar en la Figura 2.14. Se definieron variables de estados para los datos que puede ingresar el usuario, como también variables para que el componente sepa cuando mostrar los modales usados para crear las visualizaciones de distintas funciones.

El componente contiene varias funciones, por una razón de visualización y el número de líneas de código se decidió mostrar algunas funcionalidades que puedan ser representativas para el resto, para mejorar la comprensión se ocupó una notación para definir las funciones, algunas empiezan con el carácter guión bajo (`_`) para representar que son las funciones encargadas de crear un componente interno del componente Extractor Personalizado, como se puede apreciar en la Figura 2.16. La función retorna un modal que contiene todos los elementos extraídos al momento de visualizar los datos al probar un nodo, por otro lado algunas empiezan sin el carácter guión bajo (`_`) para representar que son las funciones

encargadas de la manipulación de variables, como por ejemplo lo que se puede apreciar en la Figura 2.15. La función crea una copia de los nodos extractores y elimina el nodo que fue seleccionado con la opción Eliminar Nodo, para luego restablecer las variables del estado del componente Extractor Personalizado que permiten visualizar el modal con las características de un nodo a un estado vacío, en el caso de no cumplir las restricciones, envía mensajes de advertencia al usuario, por lo general estas funciones son utilizadas en los componentes internos.

El componente tiene una función render que contiene todos los elementos que se deben crear cuando el componente Extractor Personalizado es convocado, como se puede apreciar en la Figura 2.17. El render del componente posee varios componentes internos y de utilidades, que se deben incorporar como retorno de la función, estos componentes no significan que se visualizarán instantáneamente sino que dependen de las variables de estado del componente Extractor Personalizado.

Para poder importar el componente Extractor Personalizado en otros archivos, se define su exportación como se puede apreciar en la Figura 2.18. En este caso el componente Extractor Personalizado esta conectado con Redux, por lo tanto es capaz de hacer peticiones para cambiar variables de la Store principal de la aplicación y a su vez puede capturar variables de la Store, estas variables de Store por lo general cambian gracias a peticiones asincrónicas y son importantes para varios componentes de la aplicación, por eso son definidas en la Store.

```

40
41 class ExtractorPersonalizado extends Component {
42   // STATE DEL COMPONENTE
43   constructor(props) {
44     super(props);
45     this.state = {
46       url: "",
47       contenedor: "",
48       NodosExtractores: {},
49       elementosDelNodo: {},
50       contadorVisible: true,
51
52       realizandoPrueba: "",
53
54       viendoNodo: "",
55       showMostrarNodo: false,
56       showCrearNodoModal: false,
57       showGuardarNodoModal: false,
58       showEliminarNodoModal: false,
59       showGuardarExtractorModal: false,
60       showItemsExtraidos: false,
61       showItemsExtraidosExtractor: false
62     };
63   }
64

```

Figura 2.14: State del Componente Extractor Personalizado.

```

910  /**
911   * FUNCION PARA ELIMINAR NODO EXTRACTOR
912   * LO ELIMINA DEL OBJETO DEL STATE DEL COMPONENTE
913   * MUESTRA MENSAJE DE ACUERDO A CASOS
914   */
915  eliminarNodoExtractor = () => {
916    if (this.state.viendoNodo) {
917      let auxNodosExtractores = { ...this.state.NodosExtractores };
918      delete auxNodosExtractores[this.state.viendoNodo];
919      this.setState({
920        NodosExtractores: auxNodosExtractores,
921        contadorVisible: !this.state.contadorVisible,
922        elementosDelNodo: {},
923        contenedor: "",
924        showMostrarNodo: false,
925        showEliminarNodoModal: false
926      });
927      let headers = {
928        mensaje: true,
929        texto: "El Nodo ha sido Eliminado con éxito!",
930        time: 3.5
931      };
932      this.props.mostrarMensaje(headers);
933    } else {
934      let headers = {
935        mensaje: true,
936        texto: "Ha ocurrido un error al intentar Eliminar el Nodo",
937        time: 3.5
938      };
939      this.props.mostrarMensaje(headers);
940    }
941  };
942

```

Figura 2.15: Función Eliminar Nodo Extractor del Componente Extractor Personalizado.

```

1125  * MODAL ITEMS EXTRAIDOS
1126  * CREA UN MODAL Y MUESTRA LOS
1127  * RESULTADOS DEL SERVICIO
1128  * PROBAR NODO
1129  */
1130  modalItemsExtraidos() {
1131    if (this.props.extraccionPersonalizada) {
1132      if (
1133        this.props.extraccionPersonalizada.fase ===
1134        "EXTRACCION_PERSONALIZADA_EXITOSA" &&
1135        this.props.extraccionPersonalizada.datos.length > 0
1136      ) {
1137        if (this.state.showItemsExtraidos) {
1138          return (
1139            <Modal
1140              className="itemsExtraidos-modal"
1141              backdropClassName="itemsExtraidos-backdrop"
1142              show={this.state.showItemsExtraidos}
1143              onHide={this.handleHideItemsExtraidos}
1144            >
1145              <Modal.Header closeButton>
1146                <Modal.Title>
1147                  <p className="subtitulo">Elementos Extraídos </p>
1148                  <p className="normal-text">
1149                    {this.props.extraccionPersonalizada.datos.length}
1150                  </p>
1151                </Modal.Title>
1152              </Modal.Header>
1153              <Modal.Body>
1154                <ExtraccionesNodos datos={this.props.extraccionPersonalizada} />
1155              </Modal.Body>
1156            </Modal>
1157          );
1158        }
1159      }
1160    }
1161    return null;
1162  }

```

Figura 2.16: Función Visualización Resultados Probar Nodo Nodo del Componente Extractor Personalizado.

```

1404
1405 render() {
1406     /**
1407     * CREACION DE COMPONENTES INTERNOS
1408     */
1409     let BotonesSetups = this._botonesSetups();
1410     let CrearNodoModal = this._modalCrearNodo();
1411     let GuardarNodoModal = this._modalGuardarNodo();
1412     let EliminarNodoModal = this._modalEliminarNodo();
1413     let VerNodoModal = this._modalVerNodo();
1414     let NodosExtractores = this._crearNodosExtractores();
1415     let GuardarExtractorModal = this._modalGuardarExtractor();
1416
1417     // UTILIDADES
1418     let LOADER = this._Loader();
1419     let OpcionesDescarga = this._opcionesDescarga();
1420     let BotonItemsExtraidos = this._botonItemsExtraidos();
1421     let ModalItemsExtraidos = this._modalItemsExtraidos();
1422     let BotonItemsExtraidosExtractor = this._botonItemsExtraidosExtractor();
1423     let ModalItemsExtraidosExtractor = this._modalItemsExtraidosExtractor();
1424     let OpcionesDescargaExtractor = this._opcionesDescargaExtractor();
1425     return (
1426         <div>
1427             {LOADER}
1428             {OpcionesDescarga}
1429             {BotonItemsExtraidos}
1430             {ModalItemsExtraidos}
1431             {OpcionesDescargaExtractor}
1432             {BotonItemsExtraidosExtractor}
1433             {ModalItemsExtraidosExtractor}
1434             {CrearNodoModal}
1435             {GuardarNodoModal}
1436             {EliminarNodoModal}
1437             {GuardarExtractorModal}
1438             {VerNodoModal}
1439             <InformacionExtractorPersonalizado />
1440
1441             {/* SECCION EXTRACTOR PERSONALIZADO
1442              * INPUT PAGINA WEB
1443              * NODOS EXTRACTORES
1444              * BOTONES OPCIONES
1445              */}
1446             <div className="third-wireframe">...
1447         </div>
1448     );
1449 }
1450 }

```

Figura 2.17: Función Render del Componente Extractor Personalizado.

```

1499  /**
1500   * ESTE COMPONENTE RENDERISA DEPENDIENDO DE LA STORE
1501   * DEPENDE DE:
1502   * SI EXISTE O NO USUARIO
1503   * SI EXISTE O NO RESULTADOS EXTRACCION POR EL NODO
1504   * SI EXISTE O NO RESULTADOS EXTRACCION POR EL EXTRACTOR
1505   */
1506  const mapStateToProps = state => {
1507    let { extraccion } = state.ExtraccionPersonalizada;
1508    let { solicitud } = state.Usuario;
1509    let { extractorExtraccion } = state.ExtraccionExtractor;
1510    return {
1511      extraccionPersonalizada: extraccion,
1512      extraccionExtractor: extractorExtraccion,
1513      usuario: solicitud
1514    };
1515  };
1516
1517  /**
1518   * ESTE COMPONENTE PUEDE ENVIAR PETICIONES A LA STORE:
1519   * PUEDE ENVIAR:
1520   * PETICION SERVICIO PROBAR NODO
1521   * PETICION SERVICIO PROBAR EXTRACTOR
1522   * BORRAR DATOS DE PROBAR NODO
1523   * BORRAR DATOS DE PROBAR EXTRACTOR
1524   * UN MENSAJE ESTATICO
1525   * UN NUEVO MENSAJE
1526   */
1527  function matchDispatchToProps(dispatch) {
1528    return bindActionCreators(
1529      {
1530        petitionExtraccionPersonalizada: petitionExtraccionPersonalizada,
1531        borrarDatosExtraccionPersonalizada: ExtraccionPersonalizada,
1532        petitionExtraccionExtractor: petitionExtraccionExtractor,
1533        borrarDatosExtraccionExtractor: extraccionExtractor,
1534        mostrarMensaje: mostrarMensaje,
1535        mensajesEstatico: mensajes
1536      },
1537      dispatch
1538    );
1539  }
1540
1541  export default connect(mapStateToProps, matchDispatchToProps)(
1542    ExtractorPersonalizado
1543  );
1544

```

Figura 2.18: Export del Componente Extractor Personalizado.

2.4.1.4. Validación y Verificación

Para la realización de validación y verificación de las funciones codificadas, se trabajó con un usuario real con conocimiento del lenguaje HTML en contacto con la aplicación, se le hizo una breve descripción del contenido y servicios ofrecidos por parte de la aplicación, invitándolo a seguir una serie de test unitarios, en esta oportunidad se especificaron los test relacionados con el Servicio de Crear un Extractor Personalizado:

T1. Crear Elemento Extractor:

En la Tabla 2.1, se detalla los datos ingresados para el test unitario de la función Crear Elemento Extractor, este test se puede realizar durante la creación o modificación de un nodo extractor, los datos principales para este test son la etiqueta y ruta.

Tabla 2.1: Test Unitario para función Crear Elemento Extractor.

Variables	Valores	Salida	Estado
Etiqueta	Noticia	Creación de elemento extractor exitoso	Proceso Exitoso
Ruta	a		
Elementos Extractores	Sin Elementos		
Etiqueta	Noticia	Mensaje de Advertencia : La etiqueta ya existe	Proceso Fallido
Ruta	span		
Elementos Extractores	Noticia		
Etiqueta	Not	Mensaje de Advertencia : Etiqueta debe ser \geq a 4 caracteres	Proceso Fallido
Ruta	span		
Elementos Extractores	Noticia		

T2. Eliminar Elemento Extractor:

En la Tabla 2.2, se detalla los datos ingresados para el test unitario de la función Eliminar Elemento Extractor, este test se puede realizar durante la creación o modificación de un nodo extractor, se debe haber creado al menos 1 elemento extractor, los datos principales para este test son la etiqueta del elemento extractor que se desea eliminar.

Tabla 2.2: Test Unitario para función Eliminar Elemento Extractor.

Variables	Valores	Salida	Estado
Etiqueta	Noticia	Eliminación de elemento extractor exitosa	Proceso
Elementos Extractores	Noticia		Exitoso

T3. Crear Nodo Extractor:

En la Tabla 2.3, se detalla los datos ingresados para el test unitario de la función Crear Nodo Extractor, este test se puede realizar durante la creación o modificación de un extractor, los datos principales para este test son el nombre del nodo, una breve descripción, el contenedor y los elementos extractor pertenecientes al nodo.

Tabla 2.3: Test Unitario para función Crear Nodo Extractor.

Variables	Valores	Salida	Estado
Nombre del Nodo	Noticias Ultimo Minuto	Creación del nodo extractor exitosa	Proceso Exitoso
Descripción	Extrae todo el párrafo		
Contenedor	.lista_mas_visto > li		
Elementos Extractores	Noticia		
Nodos Extractores	sin nodos		
Nombre del Nodo	Noticias Ultimo Minuto	Mensaje de Advertencia: El nombre del nodo ya existe	Proceso Fallido
Descripción	Extrae todo el párrafo		
Contenedor	.lista_mas_visto > li		
Elementos Extractores	Noticia		
Nodos Extractores	Noticias Ultimo Minuto		

T4. Probar Nodo Extractor:

En la Tabla 2.4, se detalla los datos ingresados para el test unitario de la función Probar Nodo Extractor, efectuando el test con el servicio en actividad y sin actividad, los datos principales para este test es el ingreso de una dirección Url de página web y las características del nodo que se desea probar para realizar la extracción de los textos mediante el Servicio de Probar un Nodo Extractor.

Advertencia:

Este test puede contener errores ya que la dirección Url de página ingresada puede sufrir modificaciones en su estructura y maquetación lo que puede alterar la salida y el estado de este test unitario.

Tabla 2.4: Test Unitario para función Probar Nodo Extractor.

Variables	Valores	Salida	Estado
Prueba con Servicios Activados			
Url	http://www.latercera.com	Aparecen botones para Visualizar o Descargar los resultados del nodo.	Proceso Exitoso
Nombre del Nodo	Noticias Ultimo Minuto		
Descripción	Extrae todo el párrafo		
Contenedor	.lista_mas_visto > li		
Elementos Extractores	Noticia		
Prueba con Servicios No Activados			
Url	http://www.latercera.com	Mensaje de Advertencia: Falla al conectar con el Servicio Probar Nodo.	Proceso Fallido
Nombre del Nodo	Noticias Ultimo Minuto		
Descripción	Extrae todo el párrafo		
Contenedor	.lista_mas_visto > li		
Elementos Extractores	Noticia		
Prueba con Servicios Activados			
Url	http://www.latercera.cl	Mensaje de Advertencia: No se han encontrado resultados al probar nodo.	Proceso Fallido
Nombre del Nodo	Noticias Ultimo Minuto		
Descripción	Extrae todo el párrafo		
Contenedor	.lista_mas_visto > li		
Elementos Extractores	Noticia		

T5. Editar Nodo Extractor:

En la Tabla 2.5, se detalla los datos ingresados para el test unitario de la función Editar Nodo Extractor, este test se puede realizar luego de haber creado y guardado al menos 1 nodo extractor, posteriormente se da click sobre el nodo que se desea modificar para visualizar sus características, terminando con dar click a la opción Guardar Nodo, los datos principales para este test son los datos modificados de un nodo extractor ya existente.

Tabla 2.5: Test Unitario para función Editar Nodo Extractor.

Variables	Valores	Salida	Estado
Nombre del Nodo	Noticias Ultimo Minuto	Modificación del nodo extractor exitosa	Proceso Exitoso
Descripción	Extrae noticias de lo ultimo		
Contenedor	.lista_mas_visto > li		
Elementos Extractores	Noticia		
Nodos Extractores	Noticias Ultimo Minuto		
Nombre del Nodo	Noticias Ultimo Minuto (Modificar a) Ultimo Minuto	Mensaje de Advertencia: El nuevo nombre del nodo ya existe	Proceso Fallido
Descripción	Extrae noticias de lo ultimo		
Contenedor	.lista_mas_visto > li		
Elementos Extractores	Noticia		
Nodos Extractores	Noticias Ultimo Minuto, Ultimo Minuto,		
Nombre del Nodo	Noticias Ultimo Minuto	Mensaje de Advertencia: El nodo debe contener al menos 1 elemento extractor.	Proceso Fallido
Descripción	Extrae noticias de lo ultimo		
Contenedor	.lista_mas_visto > li		
Elementos Extractores	sin elementos		
Nodos Extractores	Noticias Ultimo Minuto		

T6. Guardar Nodo Extractor:

En la Tabla 2.6, se detalla los datos ingresados para el test unitario de la función Guardar Nodo Extractor, este test se puede realizar luego de haber ingresado y creado las características requeridas para un nodo, terminando con dar click a la opción Guardar Nodo, los datos principales para este test son el nombre del nodo, una breve descripción, el contenedor y los elementos extractores pertenecientes al nodo.

Tabla 2.6: Test Unitario para función Guardar Nodo Extractor.

Variables	Valores	Salida	Estado
Nombre del Nodo	Noticias Ultimo Minuto	Guardar nodo extractor exitoso	Proceso Exitoso
Descripción	Extrae noticias de lo ultimo		
Contenedor	.lista_mas_visto > li		
Elementos Extractores	Noticia		
Nodos Extractores	sin nodos		
Nombre del Nodo	Noticias Ultimo Minuto	Mensaje de Advertencia: El nombre del nodo ya existe	Proceso Fallido
Descripción	Extrae noticias de lo ultimo		
Contenedor	.lista_mas_visto > li		
Elementos Extractores	Noticia		
Nodos Extractores	Noticias Ultimo Minuto		
Nombre del Nodo	Noticias Ultimo Minuto	Mensaje de Advertencia: El nodo debe contener al menos 1 elemento extractor.	Proceso Fallido
Descripción	Extrae noticias de lo ultimo		
Contenedor	.lista_mas_visto > li		
Elementos Extractores	sin elementos		
Nodos Extractores	Noticias Ultimo Minuto		

T7. Eliminar Nodo Extractor:

En la Tabla 2.7, se detalla los datos ingresados para el test unitario de la función Eliminar Nodo Extractor, este test se puede realizar luego de haber creado y guardado al menos 1 nodo, posteriormente se da click sobre el nodo que se desea eliminar para visualizar sus características, terminando con dar click a la opción Eliminar Nodo, los datos principales para este test es el nombre del nodo que se desea eliminar.

Tabla 2.7: Test Unitario para función Eliminar Nodo Extractor.

Variabes	Valores	Salida	Estado
Nombre del Nodo	Noticias Ultimo Minuto	Eliminación de nodo	Proceso
Nodos Extractores	Noticias Ultimo Minuto	extractor exitosa	Exitoso

T8. Crear Extractor Personalizado:

En la Tabla 2.8, se detalla los datos ingresados para el test unitario de la función Crear Extractor Personalizado, este test se puede realizar durante el uso del Servicio Crear Extractor Personalizado, los datos principales para este test son el usuario, el nombre del extractor, una breve descripción, una dirección Url de página web y los nodos extractores pertenecientes al extractor.

Tabla 2.8: Test Unitario para función Crear Extractor Personalizado.

Variabes	Valores	Salida	Estado
Usuario	usuarioprueba	Creación del extractor exitosa	Proceso Exitoso
Nombre del Extractor	La Tercera		
Descripción	Extrae noticias ultimo minuto		
Url	http://www.latercera.com		
Nodos Extractores	Noticias Ultimo Minuto		
Mis Extractores	sin extractores		
Usuario	usuarioprueba	Mensaje de Advertencia: El nombre del extractor ya existe	Proceso Fallido
Nombre del Extractor	La Tercera		
Descripción	Extrae noticias ultimo minuto		
Url	http://www.latercera.com		
Nodos Extractores	Noticias Ultimo Minuto		
Mis Extractores	La Tercera		
Usuario	usuarioprueba	Mensaje de Advertencia: El extractor debe contener al menos 1 nodo extractor.	Proceso Fallido
Nombre del Extractor	La Tercera		
Descripción	Extrae noticias ultimo minuto		
Url	http://www.latercera.com		
Nodos Extractores	sin nodos		
Mis Extractores	sin extractores		

T9. Guardar Extractor Personalizado:

En la Tabla 2.9, se detalla los datos ingresados para el test unitario de la función Guardar Extractor Personalizado, este test se puede realizar luego de haber ingresado y creado todas las características necesarias de un extractor, terminando con dar click a la opción Guardar Extractor, los datos principales para este test son el usuario que creo el extractor, nombre del extractor, una breve descripción, una dirección Url de página web y los nodos extractores pertenecientes al extractor que se desea guardar.

Tabla 2.9: Test Unitario para función Guardar Extractor Personalizado.

Variables	Valores	Salida	Estado
Usuario	usuarioprueba	Guardar características del extractor exitosa	Proceso Exitoso
Nombre del Extractor	La Tercera		
Descripción	Extrae noticias ultimo minuto		
Url	http://www.latercera.com		
Nodos Extractores	Noticias Ultimo Minuto		
Mis Extractores	sin extractores		
Usuario	usuarioprueba	Mensaje de Advertencia: El nombre del extractor ya existe	Proceso Fallido
Nombre del Extractor	La Tercera		
Descripción	Extrae noticias ultimo minuto		
Url	http://www.latercera.com		
Nodos Extractores	Noticias Ultimo Minuto		
Mis Extractores	La Tercera		
Usuario	usuarioprueba	Mensaje de Advertencia: El extractor debe contener al menos 1 nodo extractor.	Proceso Fallido
Nombre del Extractor	La Tercera		
Descripción	Extrae noticias ultimo minuto		
Url	http://www.latercera.com		
Nodos Extractores	sin nodos		
Mis Extractores	sin extractores		

T10. Probar Extractor Personalizado:

En la Tabla 2.10, se detalla los datos ingresados para el test unitario de la función Probar Extractor Personalizado, efectuando el test con el servicio en actividad y sin actividad, los datos principales para este test es el ingreso del usuario quien desea probar el extractor, el nombre del extractor, una breve descripción, una dirección Url de página web y los nodos extractores pertenecientes al extractor que se desea probar para realizar la extracción de los textos mediante el Servicio de Probar un Extractor.

Advertencia:

Este test puede contener errores ya que la dirección Url de página ingresada puede sufrir modificaciones en su estructura y maquetación lo que puede alterar la salida y el estado de este test unitario.

Tabla 2.10: Test Unitario para función Probar Extractor Personalizado.

Variabes	Valores	Salida	Estado
Prueba con Servicios Activados		Aparecen botones para Visualizar o Descargar los resultados del extractor	Proceso Exitoso
Usuario	usuarioprueba		
Nombre del Extractor	La Tercera		
Descripción	Extrae noticias ultimo minuto		
Url	http://www.latercera.com		
Nodos Extractores	Noticias Ultimo Minuto		
Prueba con Servicios No Activados		Mensaje de Advertencia: Falla al conectar con el Servicio Probar Extractor.	Proceso Fallido
Usuario	usuarioprueba		
Nombre del Extractor	La Tercera		
Descripción	Extrae noticias ultimo minuto		
Url	http://www.latercera.com		
Nodos Extractores	Noticias Ultimo Minuto		
Prueba con Servicios Activados		Mensaje de Advertencia: No se han encontrado resultados al probar extractor.	Proceso Fallido
Usuario	usuarioprueba		
Nombre del Extractor	La Tercera		
Descripción	Extrae noticias ultimo minuto		
Url	http://www.latercera.cl		
Nodos Extractores	Noticias Ultimo Minuto		

T11. Descargar Datos Extraídos por Nodos:

En la Tabla 2.11, se detalla los datos ingresados para el test unitario de la función Descargar Datos Extraídos por Nodos, este test se puede realizar luego de haber ejecutado exitosamente el Servicio de Probar Nodo Extractor y dar click al botón que hace referencia a elegir el formato de descarga de los datos extraídos por los nodos, los datos principales para este test son los datos extraídos exitosamente desde el Servicio de Probar Nodo Extractor.

Tabla 2.11: Test Unitario para función Descargar Datos Extraídos por Nodos.

Variables	Valores	Salida	Estado
Opción Archivo de Texto		Genera el archivo de Texto mostrando inicio de descarga con los resultados del nodo.	Proceso Exitoso
Nombre del Nodo	Noticias Ultimo Minuto		
Elementos Extractores	Noticias		
Resultados	resultados del nodo		
Opción Archivo CSV		Genera el archivo CSV mostrando inicio de descarga con los resultados del nodo.	Proceso Exitoso
Nombre del Nodo	Noticias Ultimo Minuto		
Elementos Extractores	Noticias		
Resultados	resultados del nodo		

T12. Visualizar Datos Extraídos por Nodos:

En la Tabla 2.12, se detalla los datos ingresados para el test unitario de la función Visualizar Datos Extraídos por Nodos, este test se puede realizar luego de haber ejecutado exitosamente el Servicio de Probar Nodo Extractor y dar click al botón que hace referencia a visualizar los datos extraídos por los nodos, los datos principales para este test son los datos extraídos exitosamente desde el Servicio de Probar Nodo Extractor.

Tabla 2.12: Test Unitario para función Visualizar Datos Extraídos por Nodos.

Variables	Valores	Salida	Estado
Nombre del Nodo	Noticias Ultimo Minuto	Muestra modal con los resultados de probar nodo clasificados por los elementos extractores.	Proceso Exitoso
Elementos Extractores	Noticias		
Resultados	resultados del nodo		

T13. Descargar Datos Extraídos por Extractores:

En la Tabla 2.13, se detalla los datos ingresados para el test unitario de la función Descargar Datos Extraídos por Extractores, este test se puede realizar luego de haber ejecutado exitosamente el Servicio de Probar Extractor y dar click al botón que hace referencia a elegir el formato de descarga de los datos extraídos por los extractores, los datos principales para este test son los datos extraídos exitosamente desde el Servicio de Probar Extractor.

Tabla 2.13: Test Unitario para función Descargar Datos Extraídos por Extractores.

Variables	Valores	Salida	Estado
Opción Archivo de Texto		Genera el archivo de Texto mostrando inicio de descarga con los resultados del extractor.	Proceso Exitoso
Nombre del Extractor	La Tercera		
Nodos Extractores	Noticias Ultimo Minuto		
Resultados	resultados del extractor		
Opción Archivo CSV		Genera el archivo CSV mostrando inicio de descarga con los resultados del extractor.	Proceso Exitoso
Nombre del Extractor	La Tercera		
Nodos Extractores	Noticias Ultimo Minuto		
Resultados	resultados del extractor		

T14. Visualizar Datos Extraídos por Extractores:

En la Tabla 2.14, se detalla los datos ingresados para el test unitario de la función Visualizar Datos Extraídos por Extractores, este test se puede realizar luego de haber ejecutado exitosamente el Servicio de Probar Extractor y dar click al botón que hace referencia a visualizar los datos extraídos por los extractores, los datos principales para este test son los datos extraídos exitosamente desde el Servicio de Probar Extractor.

Tabla 2.14: Test Unitario para función Visualizar Datos Extraídos por Extractores.

Variables	Valores	Salida	Estado
Nombre del Extractor	La Tercera	Muestra modal con los resultados de probar extractor clasificados por los nodos extractores.	Proceso Exitoso
Nodos Extractores	Noticias Ultimo Minuto		
Resultados	resultados del extractor		

2.4.2. Segundo Incremento

Para el segundo incremento se contempló el desarrollo de todo lo relacionado con el Servicio de Extractor desde Twitter y Mis Extractores.

Servicio Extractor Twitter.

El Servicio Crear Extractor Personalizado ofrece al usuario la posibilidad de ingresar una cadena de búsqueda para encontrar tweets desde Twitter, el extractor esta basado en el buscador público de Twitter (Twitter-Search, 2017), donde se admiten operadores para ser más específico en la búsqueda, el buscador no responde correctamente a operadores en español, es por esta razón que se deben ingresar en inglés, posteriormente, si se logran obtener resultados de la búsqueda el servicio ofrece visualizar o descargar los resultados obtenidos.

El Extractor de Twitter está predefinido para que capture las siguientes características de los tweets:

Nombre del Usuario, Usuario, Fecha del Tweet, Texto del Tweet, Cantidad de Comentarios, Cantidad de Retweet y Cantidad de Favoritos.

Servicio Mis Extractores.

El Servicio Mis Extractores ofrece al usuario la posibilidad de reutilizar o modificar los extractores creados y guardados desde el Servicio de Crear un Extractor Personalizado, por esta razón, al editar un extractor, se tiene el acceso a modificar todos los componentes o características que fueron necesarios en el proceso de creación, con la salvedad de que se agrega la opción de eliminar extractores.

El Servicio interactúa con la base de datos para obtener los extractores del usuario, en el caso de modificación, sólo se reflejarán cuando se guarde nuevamente a nivel del Extractor, de otro modo no se reflejarán las modificaciones en la base de datos.

2.4.2.1. Especificación de Requisitos

2.4.2.1.1 Requisitos Funcionales

A continuación se presentan los requisitos funcionales desarrollados para complementar el primer incremento de la aplicación:

- **R15. Buscar Tweets**
- **R16. Descargar Tweets Extraídos**
- **R17. Visualizar Tweets Extraídos**
- **R18. Cargar Mis Extractores**
- **R19. Eliminar Extractor Personalizado**
- **R20. Editar Extractor Personalizado**
- **R21. Registrar Usuario**
- **R22. Ingresar Usuario**
- **R23. Cambiar Contraseña**

R15. Buscar Tweets

- **Descripción:**

Permite comunicarse con el Servicio de Extracción de Twitter, obteniendo como resultados tweets relacionados con una cadena de texto ingresada por el usuario.

- **Entradas y Restricciones:**

Cadena de texto (String 1-100 caracteres obligatorio).

- **Proceso:**

R15.1: Se ingresa la cadena de texto a partir de la cual se inicia el proceso del Servicio de Extractor desde Twitter a través de la tecla ENTER.

R15.2: Si la extracción resulta exitosa, aparecerán botones para visualizar o descargar los tweets extraídos, de lo contrario se muestra un mensaje de advertencia: 'Ha ocurrido un error al ejecutar el extractor de twitter, no se han encontrado resultados, revisa los parámetros, luego, intenta nuevamente '.

R15.3: En el caso de error de conexión con el servicio, se muestra el mensaje de advertencia: 'Ha ocurrido un error al ejecutar el extractor de twitter, no se puede conectar con el Servicio de Extracción de Twitter'.

- **Salida:**

Tweets extraídos o mensaje de advertencia.

R16. Descargar Tweets Extraídos

- **Descripción:**

Da la opción al usuario de poder almacenar los tweets extraídos por el Extractor de Twitter en formato Texto y CSV.

- **Entradas y Restricciones:**

Tweets (Array of Objects 0-500 objetos obligatorio).

- **Proceso:**

R16.1: Se ingresa los tweets extraídos al usar el Servicio del Extractor de Twitter, luego se abrirá una visualización donde se da la opción al usuario de seleccionar el formato de descarga de los resultados, puede seleccionar entre Texto o CSV, posteriormente se generará el archivo correspondiente a la opción seleccionada, iniciando la descarga para el usuario.

- **Salida:**

Archivo.txt o Archivo.csv.

R17. Visualizar Tweets Extraídos

- **Descripción:**

Da la opción al usuario de poder observar cuáles fueron los tweets extraídos por el Extractor de Twitter.

- **Entradas y Restricciones:**

Tweets (Array of Objects 0-500 objetos obligatorio).

- **Proceso:**

R17.1: Se ingresa los tweets extraídos al usar el Servicio del Extractor de Twitter, luego se abrirá una visualización donde se despliegan todos tweets mostrando todas sus características.

- **Salida:**

Tweets extraídos.

R18. Cargar Mis Extractores

- **Descripción:**

Rescata la información de todos los extractores creados por un usuario en particular desde la base de datos.

- **Entradas y Restricciones:**

Usuario (String 4-50 caracteres obligatorio).

- **Proceso:**

R18.1: Se ingresa el usuario para luego iniciar una búsqueda para capturar todos los extractores que ha creado y guardado el usuario en la base de datos, posteriormente se crea una visualización para listar todos los extractores encontrados.

R18.2: En el caso de no encontrar extractores pertenecientes al usuario, se dará información para que el usuario se dirija a crear extractores mediante el Servicio de Crear Extractor Personalizado.

- **Salida:**

Extractores del usuario o recomendación de uso de Servicio Crear Extractor Personalizado.

R19. Eliminar Extractor Personalizado

- **Descripción:**

Permite borrar toda la información asociada a un extractor de un usuario en particular en la base de datos.

- **Entradas y Restricciones:**

Usuario (String 4-50 caracteres obligatorio).

Nombre del Extractor (String 4-50 caracteres obligatorio).

- **Proceso:**

R19.1: Se selecciona un extractor desde la visualización de los extractores a través de clicar sobre un extractor, posteriormente se abrirá una visualización con todas las características del extractor seleccionado, luego se inicia la eliminación a través del botón Eliminar Extractor, el cual abrirá una confirmación de eliminación, en el caso de aceptar la confirmación se procederá a eliminar los registros del extractor seleccionado en los extractores perteneciente al usuario en la base de datos, mostrando el mensaje de éxito: 'Se ha eliminado el extractor con éxito!', en el caso de rechazar la confirmación se dará por terminado el proceso.

- **Salida:**

Eliminación de extractor exitosa.

R20. Editar Extractor Personalizado

■ Descripción:

Permite almacenar en la base de datos las características de un extractor existente que ha sufrido modificaciones, asociándolo al usuario, otorgando la posibilidad de ser ocupado nuevamente mediante el Servicio Mis Extractores.

■ Entradas y Restricciones:

Usuario (String 4-50 caracteres obligatorio).

Url (String 16-255 caracteres obligatorio).

Nombre del Extractor (String 4-50 caracteres obligatorio).

Descripción (String 4-255 caracteres obligatorio).

Nodos Extractores (Object 1-500 keys-values obligatorio).

■ Proceso:

R20.1: Se selecciona un extractor desde la visualización de los extractores a través de clicar sobre un extractor, posteriormente se abrirá una visualización con todas las características del extractor seleccionado dando la oportunidad de modificarlas, luego de modificar, se inicia la modificación de las características del extractor a través del botón Guardar Extractor.

R20.2: Si el nombre del extractor modificado es único, se modifican las características en el extractor seleccionado a partir de la visualización de los extractores mostrando el mensaje de éxito: 'Se ha guardado el extractor con éxito!.', de lo contrario se muestra el mensaje de advertencia: 'Error al guardar el extractor, el nombre del extractor ya existe.'.

R20.3: En el caso de guardar las modificaciones pero estas no poseen nodos extractores, se muestra el mensaje de advertencia: 'El extractor debe tener al menos 1 nodo extractor.'

■ **Salida:**

Modificaciones del extractor exitosa o mensaje de advertencia.

R21. Registrar Usuario

■ Descripción:

Permite almacenar información de un nuevo usuario en la base de datos para posteriormente ingresar a la plataforma, obteniendo los privilegios de tener acceso a los Servicios .

■ Entradas y Restricciones:

Correo electrónico (String 8-50 caracteres obligatorio).

Usuario (String 4-50 caracteres obligatorio).

Contraseña (String 8-30 caracteres obligatorio).

Primer Nombre (String 4-50 caracteres obligatorio).

Segundo Nombre (String 4-50 caracteres obligatorio).

Primer Apellido (String 4-50 caracteres obligatorio).

Segundo Apellido (String 4-50 caracteres obligatorio).

Pista para Palabra Clave (String 4-50 caracteres obligatorio).

Palabra Clave (String 4-50 caracteres obligatorio).

■ Proceso:

R21.1: Se ingresan los datos requeridos para esta función, de los cuales se inicia el proceso de registro para un usuario a través del botón Completar Registro.

R21.2: Si el correo electrónico y el usuario son únicos, se guardarán los datos como un nuevo usuario en la base de datos mostrando el mensaje de éxito: 'Registro Exitoso!, ya puedes ingresar a la plataforma.', de lo contrario se muestra el mensaje de advertencia: 'Error en el registro, el usuario o el correo electrónico ya existen.'

■ Salida:

Registro de usuario exitoso o mensaje de advertencia.

R22. Ingresar Usuario

- **Descripción:**

Permite verificar si el usuario pertenece a la base de datos para posteriormente dar los privilegios de acceso a los Servicios de la aplicación.

- **Entradas y Restricciones:**

Usuario (String 4-50 caracteres obligatorio).

Contraseña (String 8-30 caracteres obligatorio).

- **Proceso:**

R22.1: Se ingresa el usuario y contraseña iniciando el intento de ingreso a la plataforma a través del botón Ingresar, para luego comparar el usuario y contraseña con los datos de usuarios ya registrados en la base de datos.

R22.2: Si el intento de ingreso a la plataforma resulta exitoso, se dará los privilegios para que pueda ocupar los Servicios de la aplicación mostrando el mensaje de éxito: 'Bienvenido! Nombre Completo del Usuario .', de lo contrario se muestra el mensaje de advertencia: 'Usuario o Contraseña Incorrectas.'

- **Salida:**

Ingreso a la plataforma exitoso o mensaje de advertencia.

R23. Cambiar Contraseña

- **Descripción:**

Permite modificar la contraseña de un usuario mediante una palabra clave .

- **Entradas y Restricciones:**

Correo electrónico (String 8-50 caracteres obligatorio).

Usuario (String 4-50 caracteres obligatorio).

Pista para Palabra Clave (String 4-50 caracteres obligatorio).

Palabra Clave (String 4-50 caracteres obligatorio).

Nueva Contraseña (String 8-30 caracteres obligatorio).

- **Proceso:**

R23.1: Se ingresa el correo electrónico y el usuario para pedir los datos relacionados al usuario en la base de datos a través del botón Enviar Datos, luego se muestra la pista para la palabra clave recuperada desde la base de datos para ayudar al usuario a ingresar la palabra clave que ingresada en el proceso de registro, completando el ingreso a través del botón Enviar Palabra Clave, posteriormente se le da la oportunidad al usuario de ingresar una nueva contraseña, completando el cambio de contraseña a través del botón Cambiar Contraseña.

R23.2: Si el correo electrónico y el usuario no son encontrados en la base de datos, se muestra el mensaje de advertencia: 'Usuario o Correo Electrónico incorrectos.'.

R23.3: Si la palabra clave no concuerda con la almacenada en la base de datos, se muestra el mensaje de advertencia: 'Palabra Clave incorrecta .'.
..

R23.4: Si el cambio de contraseña resulta exitoso se muestra el mensaje de éxito: 'Se ha cambiado la contraseña exitosamente!.'

■ **Salida:**

Cambio de contraseña exitoso o mensaje de advertencia.

2.4.2.1.2 Requisitos no Funcionales

A continuación se presentan los requisitos no funcionales considerados para complementar el primer incremento en el desarrollo de la aplicación:

- **Portabilidad:**

- El sistema debe funcionar como mínimo en los navegadores web Google Chrome versión 64.0.3 y Mozilla Firefox versión 58.0.1.

- **Seguridad:**

- El sistema debe estar estructurado considerando cuentas de usuarios.
- El sistema debe como mínimo encryptar las contraseñas de los usuario, para el resguardo de la privacidad.

- **Integridad de la Información:**

- El sistema debe impedir la inconsistencia de la información, tanto del usuario como la de sus extractores, es decir, que un usuario no pueda borrar o alterar información de otro usuario.

2.4.2.2. Diseño de Interfaz y Modelo de Datos

2.4.2.2.1 Mockups

A continuación se presentan los Mockups desarrollados para complementar el primer incremento de la aplicación:

Se decidió no crear Mockups para cada funcionalidad, por lo tanto se desarrollaron las funcionalidades más importantes y que aportaban más contenido a la aplicación.

- **I6. Extractor de Twitter**
- **I7. Opciones de Formato**
- **I8. Visualización de Tweets**
- **I9. Mis Extractores**
- **I10. Editar Extractor**
- **I11. Eliminar Extractor**

I6. Extractor de Twitter

La interfaz del Extractor de Twitter representada en la Figura 2.19, es la sección de la aplicación donde se puede ingresar una cadena de texto para encontrar tweets relacionados a un tema en particular.

El primer paso es especificar la cadena de texto para posteriormente ejecutar la búsqueda (tecla ENTER), en el caso de encontrar resultados por parte del extractor, se mostrarán botones que darán la opciones al usuario de visualizar o descargar los tweets obtenidos, como se puede apreciar en la Figura 2.19.



Figura 2.19: Interfaz Extractor de Twitter.

I7. Opciones de Formato

La interfaz de Opciones de Formato representada en la Figura 2.20, es la sección de la aplicación donde se presentan las opciones de estructuración para los resultados obtenidos desde la ejecución de los extractores, ya sea en el extractor de twitter, personalizado o desde la sección de los extractores del usuario.

La diferencia de la organización o estructuración de los resultados depende del extractor, en el caso de Twitter, los resultados son clasificados en forma de tweets, en el extractor personalizado son clasificados a partir de los nodos y elementos extractores que lo componen.

El primer paso es especificar el formato (clickear), luego la aplicación generará la estructura de los resultados obtenidos acorde a la opción elegida por el usuario.



Figura 2.20: Interfaz Opciones de Formato.

18. Visualización de Tweets

La interfaz de Visualización de Tweets representada en la Figura 2.21, es la sección de la aplicación donde se muestran en forma gráfica y estructurada los resultados obtenidos por el extractor de Twitter.

Esta interfaz se tomará como base para las visualizaciones de los resultados obtenidos al probar extractores o nodos.

La diferencia de la organización o estructuración de los resultados depende del extractor, en el caso de Twitter, los resultados son clasificados en forma de tweets, en el extractor personalizado son clasificados a partir de los nodos y elementos extractores que lo componen.

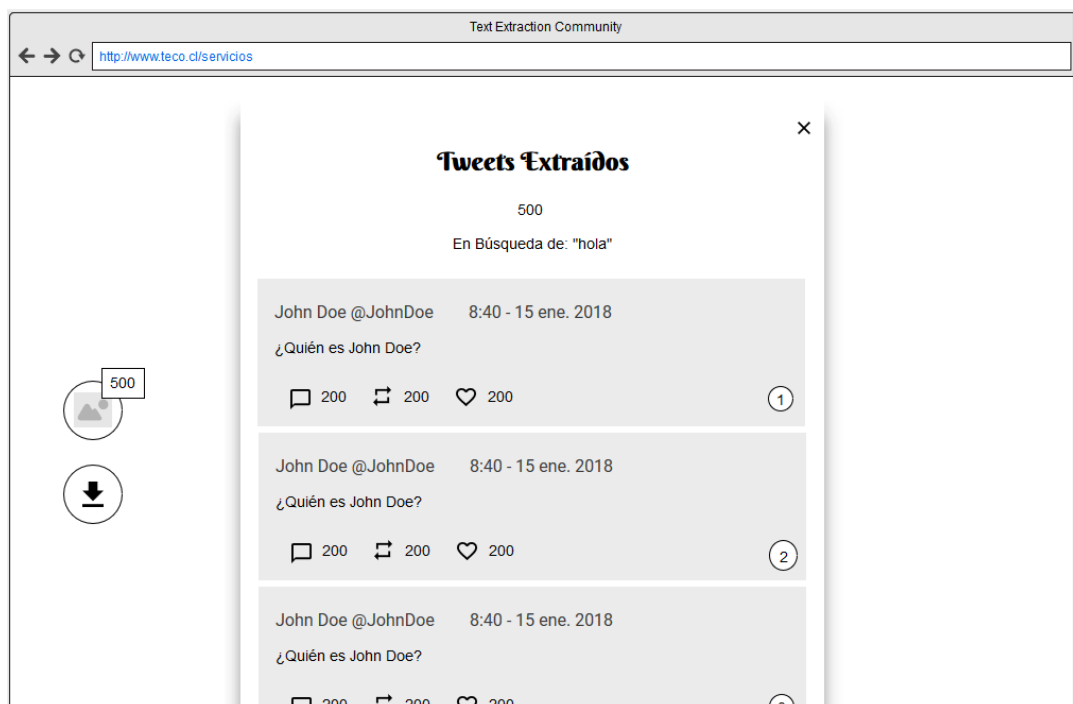


Figura 2.21: Interfaz Visualización de Resultados.

I9. Mis Extractores

La interfaz de Mis Extractores representada en la Figura 2.22, es la sección de la aplicación donde se muestran en forma gráfica y estructurada todos los extractores que ha creado y guardado el usuario.

La interfaz contiene características llamativas de los extractores, ejemplo: nombre del extractor, la página web y una descripción, como se puede apreciar en la Figura 2.22.

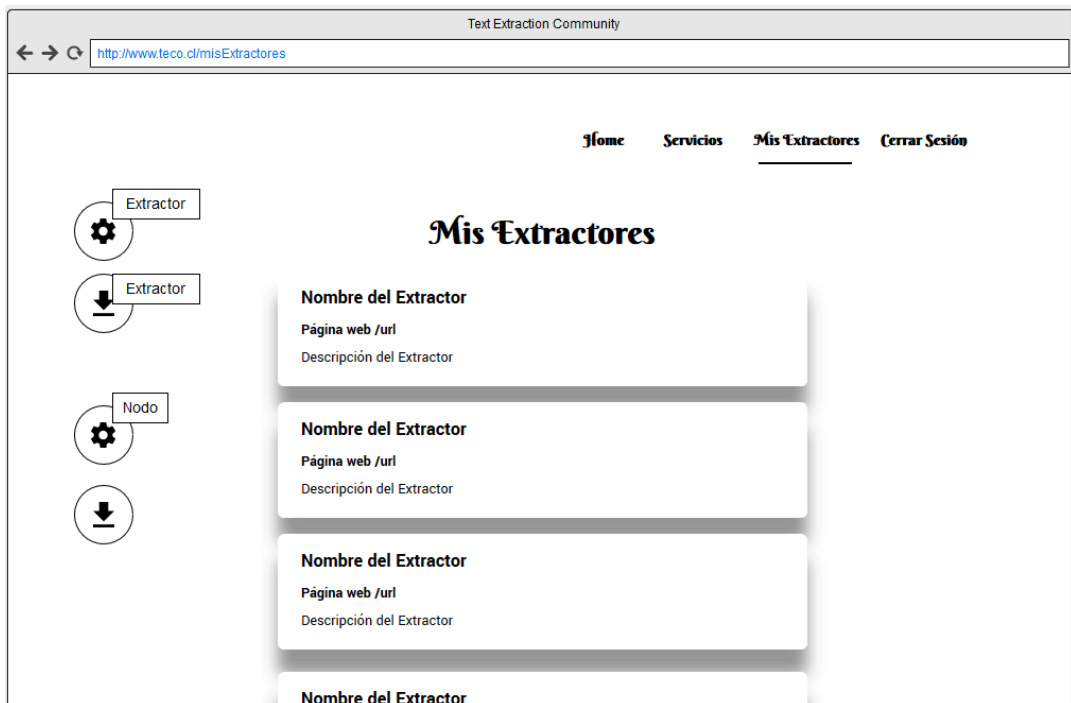
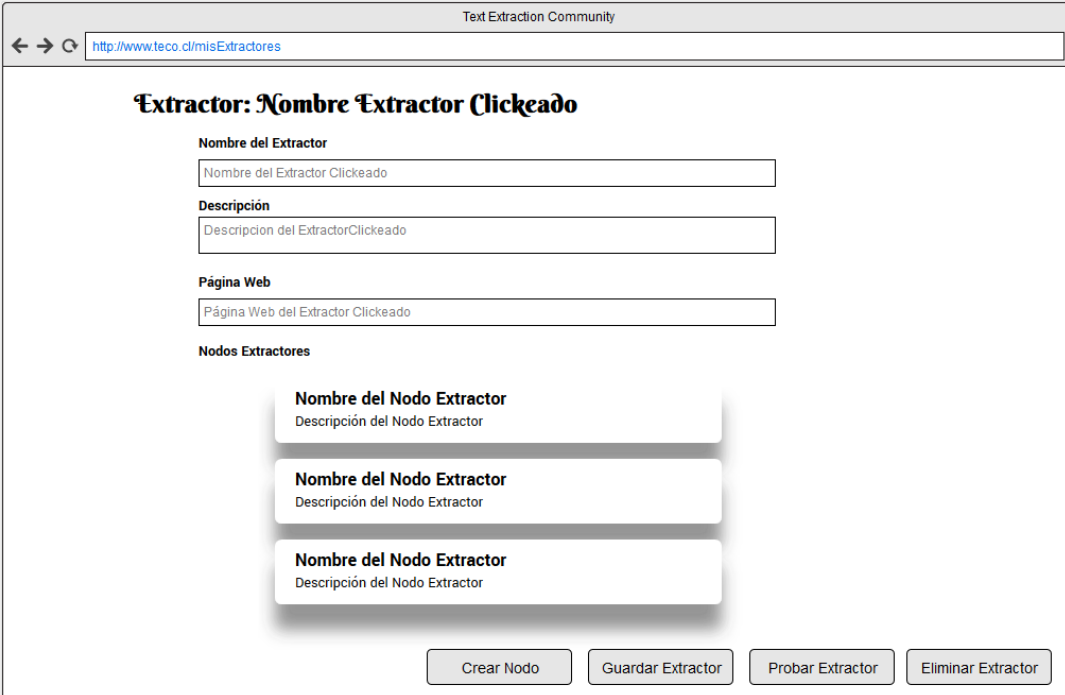


Figura 2.22: Interfaz Mis Extractores.

I10. Editar Extractor

La interfaz de Editar un Extractor representada en la Figura 2.23, es la sección de la aplicación donde se pueden modificar los componentes y características del extractor, teniendo acceso a todas las funcionalidades presentadas para la creación de un extractor personalizado, además de agregar una nueva funcionalidad para eliminar el extractor.

Para llegar a la interfaz se deberá elegir (clickear) un extractor desde la lista de extractores del usuario como se puede apreciar en la Figura 2.22.



The screenshot shows a web browser window titled "Text Extraction Community" with the URL "http://www.teco.cl/misExtractores". The main heading is "Extractor: Nombre Extractor Clickeado". Below this, there are three input fields: "Nombre del Extractor" (containing "Nombre del Extractor Clickeado"), "Descripción" (containing "Descripción del ExtractorClickeado"), and "Página Web" (containing "Página Web del Extractor Clickeado"). Under the heading "Nodos Extractores", there are three identical card-like elements, each containing "Nombre del Nodo Extractor" and "Descripción del Nodo Extractor". At the bottom, there are four buttons: "Crear Nodo", "Guardar Extractor", "Probar Extractor", and "Eliminar Extractor".

Figura 2.23: Interfaz Editar Extractor.

2.4.2.2.2 Diagrama de Actividades

A continuación se presentan los diagramas de actividades desarrollados para complementar el primer incremento de la aplicación:

- **D2. Diagrama de Actividades Inicio de la Plataforma**
- **D3. Diagrama de Actividades Extractor Twitter**
- **D4. Diagrama de Actividades Mis Extractores**

D2. Diagrama de Actividades Inicio de la Plataforma

El diagrama del Inicio de la Plataforma muestra el flujo de las actividades que se pueden ejecutar al principio de la interacción con el usuario. La importancia del diagrama radica en dar a conocer los servicios que ofrece la aplicación (Extractor de Twitter, Mis Extractores y Crear Extractor Personalizado), como se puede apreciar en la Figura 2.24. Para la utilización de los servicios es necesario ser un usuario registrado en la plataforma.

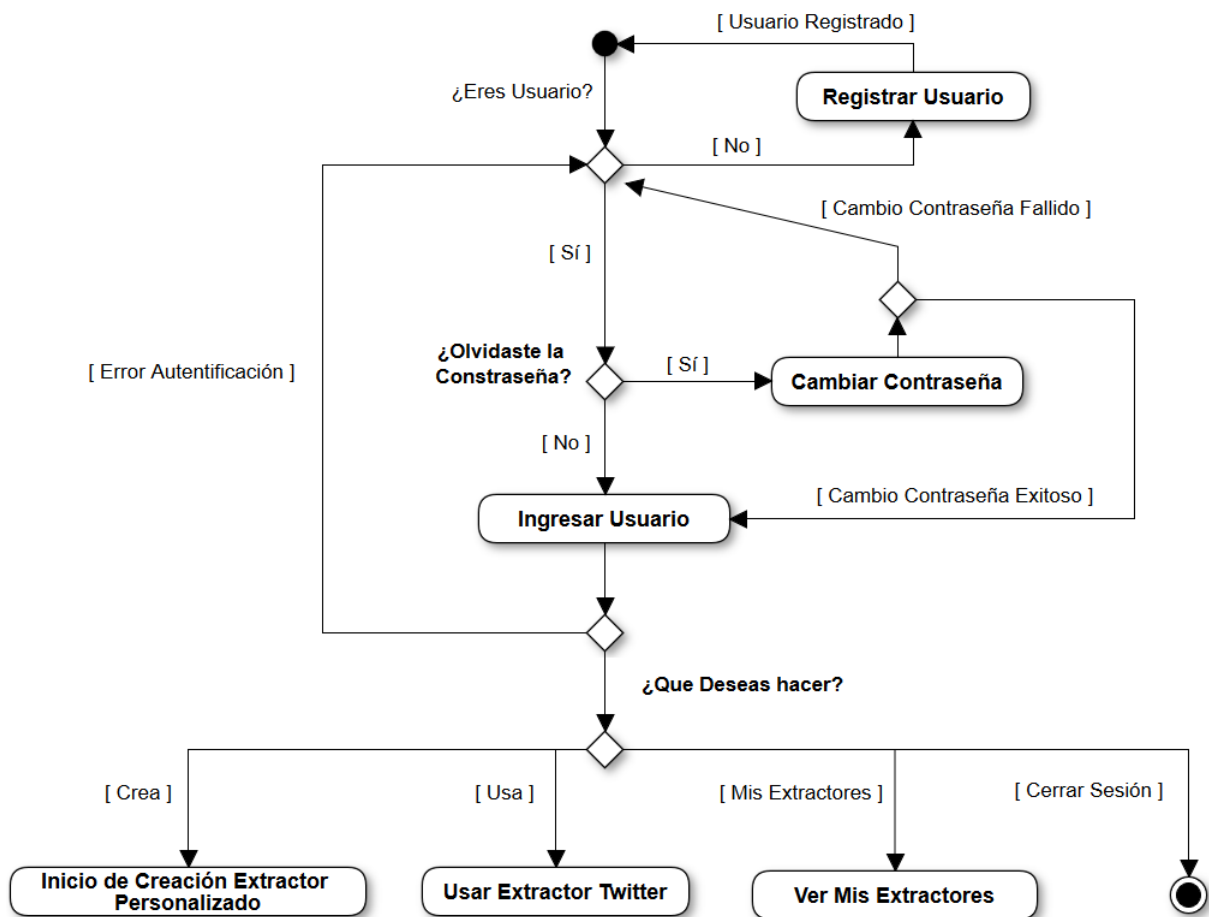


Figura 2.24: Diagrama de Actividades Inicio de la Plataforma.

D3. Diagrama de Actividades Extractor Twitter

El diagrama del Extractor de Twitter muestra el flujo de las actividades que se pueden realizar cuando se quiere usar el extractor, como se puede apreciar en la Figura 2.25. Dando la posibilidad de buscar tweets para posteriormente descargar o visualizar los resultados obtenidos.

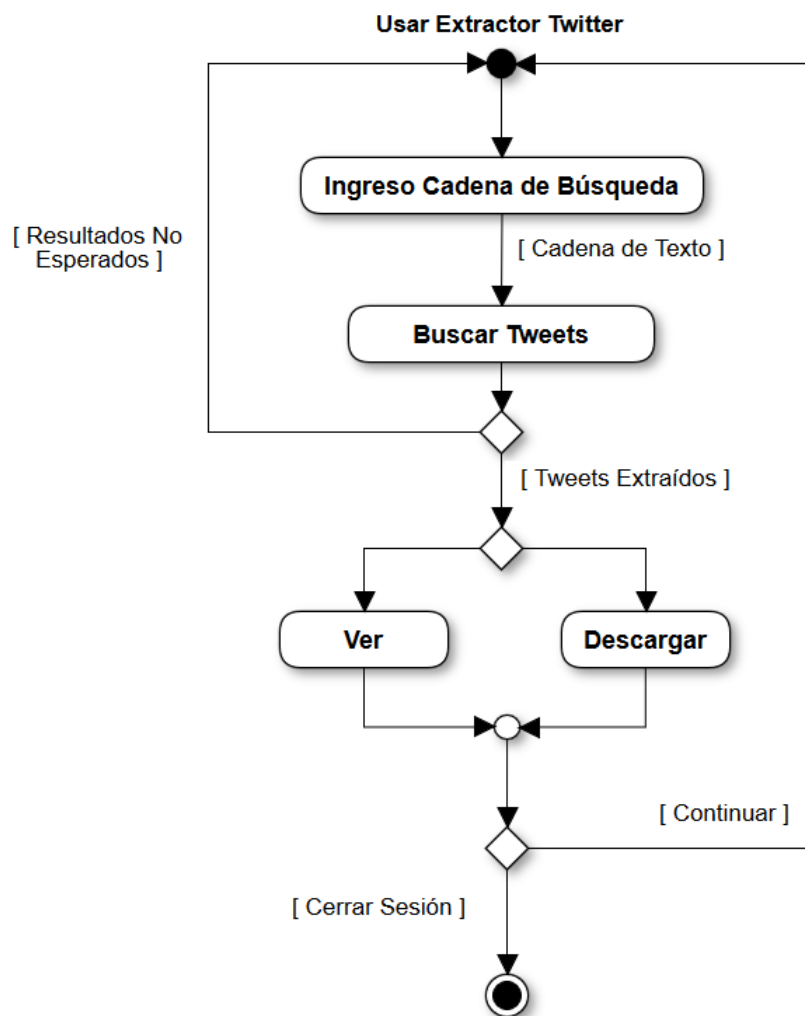


Figura 2.25: Diagrama de Actividades Extractor de Twitter.

D4. Diagrama de Actividades Mis Extractores

El diagrama de Mis Extractores representado en la Figura 2.26, muestra el flujo de las actividades que se pueden realizar cuando se desea utilizar o modificar extractores que han sido creado y guardados anteriormente por el usuario.

Lo que va detrás del servicio incide en la reutilización de procesos que forman parte del servicio Crear Extractor Personalizado, dándole la opción al usuario de modificar componentes y características del extractor.

Al estar basado en funcionalidades del servicio Crear Extractor Personalizado, la representación del diagrama de actividades Mis Extractores es similar a la Figura 2.13, con la salvedad que se agregaron nuevas funciones como lo son, editar y eliminar extractores, las cuales caracterizan y definen el servicio de reutilizar extractores creados y guardados con anterioridad.

2.4.2.2.3 Diagrama de Despliegue

El diagrama de Despliegue es una representación de la plataforma en la cual se ejecuta el software o la aplicación, a través de nodos, componentes y las relaciones o asociaciones.

En esta oportunidad, el diagrama de despliegue es usado para modelar la aplicación Text Extraction Community como un sistema Cliente-Servidor, como se puede apreciar en la Figura 2.27. Dando la oportunidad de especificar los servicios, cuáles son los componentes necesarios para la ejecución de la aplicación, como también, las funciones que requieren una interacción con la base de datos.

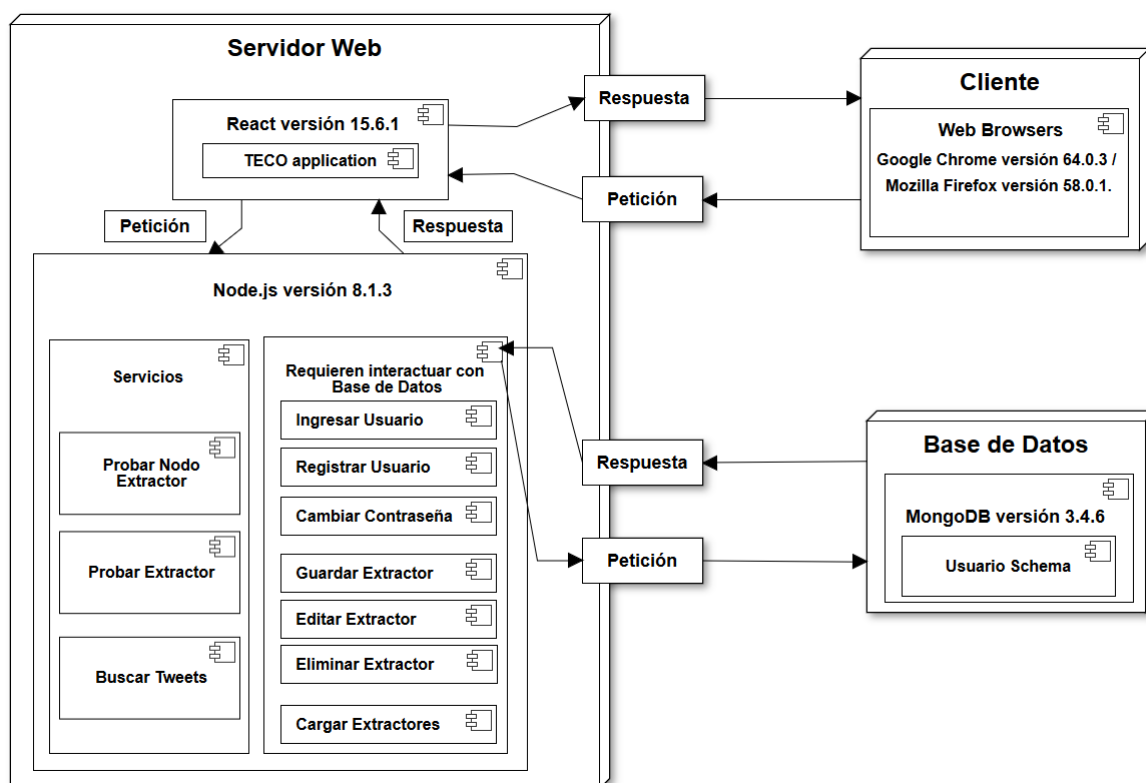


Figura 2.27: Diagrama de Despliegue Aplicación TECO.

2.4.2.2.4 Modelamiento de Datos

El Modelamiento de Datos permite describir el tipo de datos que existen en una Base de Datos y la forma en que se relacionan, para una base de datos SQL relacional, basadas en el lenguaje SQL, tablas, índices, claves primarias y foráneas, lo usual es describirla a través de diagramas Modelo Entidad Relación, Modelo Relacional, etc. En cambio para una base de datos NoSQL no relacional, no poseen diagramas para describirlas, ya que no poseen una estructura fija para los registros disminuyendo la consistencia para aumentar escalabilidad de los datos, además de no usar SQL como lenguaje principal sino que aceptan más lenguajes.

Para el desarrollo de la aplicación Text Extraction Community se utilizó MongoDB, un gestor de base de datos NoSQL basado en documentos, el cual permite uso del lenguaje JavaScript para realizar consultas, desarrollado en c++, con un sistema propio llamado BSON como una evolución de JSON con la peculiaridad de ser almacenado en código binario, aumentado la capacidad y rapidez de operaciones.

MongoDB no posee un diagrama específico para describir el modelo de sus datos, pero como utiliza JSON para interactuar con las peticiones, si se puede hacer una especificación de los datos o esquemas que se almacenarán en la base de datos, para el JSON utilizado en la aplicación Text Extraction Community se especificó la estructura representada en la Figura 2.28.

Para mejor comprensión del modelamiento de datos se realizó una adaptación del objeto JSON que se utilizó en la aplicación Text Extraction Community en un Diagrama Entidad-Relación como se puede apreciar en la Figura 2.29.

Objeto JSON:

Usuario {

<code>_id :</code>	<code>string, 24 caracteres (automática de la Base de Datos)</code>
<code>correoElectronico :</code>	<code>string, 8-50 caracteres</code>
<code>usuario :</code>	<code>string, 4-50 caracteres</code>
<code>contrasena :</code>	<code>string, 8-30 caracteres (hash encriptado 60 caracteres)</code>
<code>primerNombre :</code>	<code>string, 4-50 caracteres</code>
<code>segundoNombre :</code>	<code>string, 4-50 caracteres</code>
<code>primerApellido :</code>	<code>string, 4-50 caracteres</code>
<code>segundoApellido :</code>	<code>string, 4-50 caracteres</code>
<code>pistaPalabraClave :</code>	<code>string, 4-50 caracteres</code>
<code>palabraClave :</code>	<code>string, 4-50 caracteres</code>
extractores {	
"Nombre del Extractor" {	<code>string, 4-50 caracteres</code>
url :	<code>string, 16-255 caracteres</code>
descripcion :	<code>string, 4-255 caracteres</code>
nodosExtractores {	
"Nombre del Nodo" {	<code>string, 4-50 caracteres</code>
Descripción :	<code>string, 4-255 caracteres</code>
Contenedor :	<code>string, 4-255 caracteres</code>
ElementosExtractores {	
Etiqueta :	<code>string, 4-30 caracteres</code>
Ruta :	<code>string, 1-255 caracteres</code>
}	
}	
}	
}	
}	

Figura 2.28: Especificación JSON aplicación TECO.

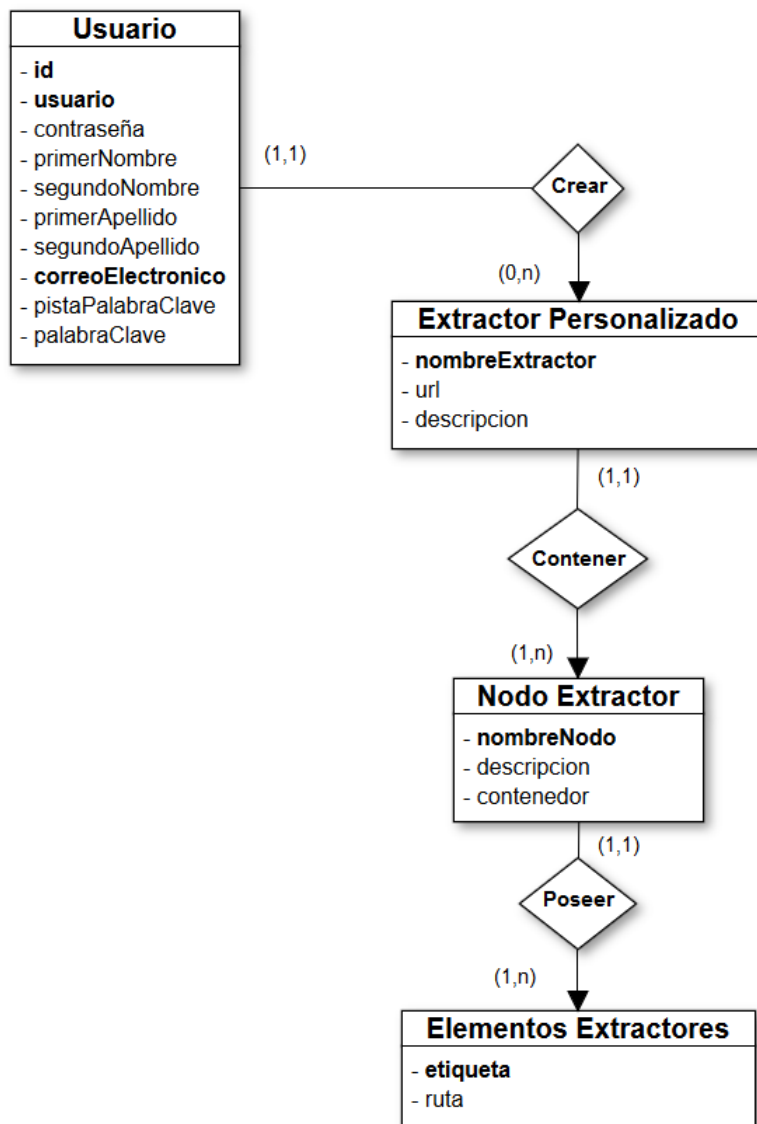


Figura 2.29: Adaptación del JSON a Entidad-Relación.

2.4.2.3. Codificación

Para el proceso de codificación en el segundo incremento se tomó como referencia las dependencias especificadas en el primer incremento para complementar con la codificación de las funciones restantes relacionadas con los Servicios Mis Extractores y Extractor de Twitter.

En esta oportunidad fue necesario crear la codificación del servidor en otra carpeta o proyecto, con el objetivo de crear los servicios correspondientes, además de las conexiones con la base de datos, para ésto fue necesario especificar las dependencias correspondientes al servidor.

Para la codificación de las funciones correspondientes al servidor se utilizó las siguientes dependencias:

Dado el número de dependencias se describieron las más importantes, en el caso de querer revisar todas las dependencias se deben dirigir al archivo package.json del servidor, el cual contiene todas las dependencias con sus respectivas versiones.

- **bcrypt (versión 1.0.3):**

Permite encriptar y desencriptar texto a través de Hashs (Github-Bcrypt, 2017).

- **body-parser (versión 1.17.2):**

Permite Manipular los datos de las peticiones HTTP que llegan al servidor a través de la propiedad req.body (Github-Body-Parser, 2017).

- **connect-mongo (versión 2.0.1):**

Session Store de MongoDB para Express (Github-Connect-Mongo, 2017).

- **cors (versión 2.8.4):**

Permite manipular las peticiones CORS del proyecto (Github-Cors, 2017).

- **express (versión 4.15.4):**

Permite configurar rutas de peticiones HTTP de la aplicación de forma pequeña y robusta (Github-Express, 2017).

- **mongoose (versión 4.11.6):**

Permite crear esquemas y modelos de objetos para MongoDB (Github-Mongoose, 2017).

- **nightmare (versión 2.10.0):**

Permite automatizar la navegación imitando las acciones de usuario, es generalmente utilizado para testing, crawling y scraping (Github-Nitghtmare, 2017).

En el Segundo Incremento se codificó las funciones relacionadas con el Servicio de Extractor de Twitter, Servicio Mis Extractores y el Servidor.

Se creó un componente para el Extractor de Twitter, donde maneja estados que le permite manipular componentes internos de acuerdo a la interacción del usuario, como se puede apreciar en la Figura 2.30. Se definieron variables de estados para los datos que puede ingresar el usuario, como también variables para que el componente sepa cuando mostrar los modales usados para crear las visualizaciones de distintas funciones.

```

16
17 class ExtractorTwitter extends Component {
18   // STATE DEL COMPONENTE
19   constructor(props) {
20     super(props);
21     this.state = { show: false };
22   }
23
24   /**
25    * MANIPULACION DE MODAL
26    * RESULTADOS DE EXTRACCION TWEETS
27    */
28   handleShow = () => {
29     this.setState({ show: true });
30   };
31
32   handleHide = () => {
33     this.setState({ show: false });
34   };
35
36   /**
37    * ENVIO DE TEXTO DE BUSQUEDA
38    * ENVIO A SERVICIO DE EXTRACTOR DE TWITTER
39    * EN EL LADO DEL SERVIDOR
40    */
41   prepararPeticion = () => {
42     this.refs.busqueda.inputRef.blur();
43     var peticion = {
44       servidor: urlDB_Servicio_Twitter,
45       busqueda: this.refs.busqueda.inputRef.value
46     };
47
48     this.props.peticionExtraccion(peticion);
49   };
50

```

Figura 2.30: State del Componente Extractor Twitter.

El componente contiene varias funciones, por una razón de visualización y el número de líneas de código se decidió mostrar algunas funcionalidades que puedan ser representativas para el resto, para mejorar la comprensión se ocupó una notación para definir las funciones, algunas empiezan con el carácter guión bajo (`_`) para representar que son las funciones encargadas de crear un componente interno del componente Extractor de Twitter, como se puede apreciar en la Figura 2.31. La función retorna un modal que contiene todos los tweets extraídos al momento de visualizar los datos al ejecutar el Servicio Extractor de Twitter, por

otro lado algunas empiezan sin el carácter guión bajo (`_`) para representar que son las funciones encargadas de la manipulación de variables, como por ejemplo lo que se puede apreciar en la Figura 2.30. La función captura la cadena de texto ingresada por el usuario creando un objeto para posteriormente delegar el envío de la petición al Servicio de Extractor desde Twitter a un action-reducer cambiando los resultados en la Store de la aplicación.

```
52  * COMPONENTE MODAL DE TWEETS EXTRAIDOS
53  * MUESTRA LA CANTIDAD DE TWEETS
54  * LA BUSQUEDA
55  * LOS TWEETS EXTRAIDOS
56  */
57  _modalTweetsExtraidos() {
58    if (this.props.extraccion) {
59      if (
60        this.props.extraccion.fase === "EXITOSA" &&
61        this.props.extraccion.datos.length > 0
62      ) {
63        if (this.state.show) {
64          return (
65            <Modal
66              className="tweetsExtraidos-modal"
67              show={this.state.show}
68              onHide={this.handleHide}
69            >
70              <Modal.Header closeButton>
71                <Modal.Title>
72                  <p className="titulo">Tweets Extraídos </p>
73                  <p className="normal-text">
74                    {this.props.extraccion.datos.length}
75                  </p>
76                  <p className="normal-text">
77                    En Búsqueda de: {this.props.extraccion.busqueda}
78                  </p>
79                </Modal.Title>
80              </Modal.Header>
81              <Modal.Body>
82                <Tweets datos={this.props.extraccion} />
83              </Modal.Body>
84            </Modal>
85          );
86        }
87      }
88    }
89    return null;
90  }
```

Figura 2.31: Componente interno Modal de Tweets Extraídos del Componente Extractor Twitter.

El componente tiene una función render que contiene todos los elementos que

se deben crear cuando el componente Extractor de Twitter es invocado, como se puede apreciar en la Figura 2.32. El render del componente posee varios componentes internos y de utilidades, que se deben incorporar como retorno de la función, estos componentes no significan que se visualizarán instantáneamente sino que dependen de las variables de estado del componente Extractor de Twitter.

```
159 render() {
160   /**
161    * CREACION DE COMPONENTES INTERNOS
162    */
163   let TweetsExtraidos = this._modalTweetsExtraidos();
164   let ButtonMostrarTweetsExtraidos = this._mostrarTweetsExtraidos();
165   let OpcionesDescarga = this._opcionesDescarga();
166   let ExtraccionLoader = this._renderisarLoader();
167   return (
168     <div>
169       {TweetsExtraidos}
170       {ButtonMostrarTweetsExtraidos}
171       {OpcionesDescarga}
172       {ExtraccionLoader}
173
174       {/**
175        * PRIMERA SECCION COMO USAR
176        */}
177       <div className="first-wireframe"> ...
178     </div>
179
193
194     {/**
195      * SEGUNDA SECCION OPERADORES DE BUSQUEDA
196      */}
197     <div className="second-wireframe"> ...
336   </div>
337
338     {/**
339      * TERCERA SECCION EXTRACTOR TWITTER
340      * INPUT PARA INGRESAR TEXTO DE BUSQUEDA
341      */}
342     <div className="third-wireframe"> ...
370   </div>
371 </div>
372 );
373 }
374 }
```

Figura 2.32: Función Render del Componente Extractor Twitter.

Para poder importar el componente Extractor de Twitter en otros archivos, se define su exportación como se puede apreciar en la Figura 2.33. En este caso el componente Extractor de Twitter esta conectado con Redux, por lo tanto es capaz de hacer peticiones para cambiar variables de la Store principal de la aplicación y a su vez puede capturar variables de la Store, estas variables de Store por lo general cambian gracias a peticiones asincrónicas y son importantes para varios componentes de la aplicación, por eso son definidas en la Store.

```
376  /**
377   * ESTE COMPONENTE RENDERISA DEPENDIENDO DE LA STORE
378   * DEPENDE DE:
379   * SI EXISTE O NO RESULTADOS DE EXTRACCION
380   */
381  const mapStateToProps = state => {
382    let { extraccion } = state.Extraccion;
383    return {
384      extraccion: extraccion
385    };
386  };
387
388  /**
389   * ESTE COMPONENTE PUEDE ENVIAR PETICIONES A LA STORE:
390   * PUEDE ENVIAR:
391   * UNA PETICION PARA EL EXTRACTOR DE TWITTER
392   */
393  function matchDispatchToProps(dispatch) {
394    return bindActionCreators(
395      {
396        petitionExtraccion: petitionExtraccion
397      },
398      dispatch
399    );
400  }
401
402  export default connect(mapStateToProps, matchDispatchToProps)(ExtractorTwitter);
403
```

Figura 2.33: Export del Componente Extractor Twitter.

Se creó un componente para el Servicio Mis Extractores, donde maneja estados que le permite manipular componentes internos de acuerdo a la interacción del usuario, como se puede apreciar en la Figura 2.34. Este componente posee variables similares a las del Extractor Personalizado representado en la Figura 2.14. La diferencia radica en que se agregan las variables para el nivel del extractor, como lo son 'viendoExtractor' y 'showMostrarExtractor'.

```
46
47 class MisExtractores extends Component {
48   // STATE DEL COMPONENTE
49   constructor(props) {
50     super(props);
51     this.state = {
52       url: "",
53       contenedor: "",
54       NodosExtractores: {},
55       elementosDelNodo: {},
56       contadorVisible: true,
57
58       realizandoPrueba: "",
59
60       viendoNodo: "",
61       showMostrarNodo: false,
62       showCrearNodoModal: false,
63       showGuardarNodoModal: false,
64       showEliminarNodoModal: false,
65       showItemsExtraidos: false,
66
67       viendoExtractor: "",
68       showMostrarExtractor: false,
69       showGuardarExtractorModal: false,
70       showEliminarExtractorModal: false,
71       showItemsExtraidosExtractor: false
72     };
73   }
```

Figura 2.34: State del Componente Mis Extractores.

El componente del Servicio Mis Extractores posee funciones similares a las del Servicio Crear Extractor Personalizado, pero la diferencia radica en que se agregan nuevas funcionalidades dado que se agregó un nivel de visualización para los extractores, estas nuevas funciones son para crear un componente de visualización de las características del extractor que sea seleccionado, guardar las modificaciones realizadas al extractor y para eliminar completamente el extractor en la base de datos, como se puede apreciar en la Figura 2.35.

```
332  /**
333  * MODAL VER EXTRACTOR
334  * CREA UN MODAL Y MUESTRA
335  * NOMBRE DEL EXTRACTOR
336  * DESCRIPCION
337  * PAGINA WEB
338  * NODOS EXTRACTORES
339  * BOTONES
340  * CREAR NODO
341  * GUARDAR EXTRACTOR
342  * PROBAR EXTRACTOR
343  * ELIMINAR EXTRACTOR
344  */
345  modalVerExtractor() { ...
490  }
491
492  /**
493  * FUNCION PARA EDITAR EL EXTRACTOR
494  * SE ENVIA DATOS A LA BASE DE DATOS
495  * USUARIO
496  * NOMBRE ANTIGUO DEL EXTRACTOR
497  * NOMBRE DEL EXTRACTOR
498  * DATOS DEL EXTRACTOR
499  * URL - DESCRIPCION - NODOS EXTRACTORES
500  * MUESTRA MENSAJES DE ACUERDO A CASOS
501  */
502  editarExtractor = () => { ...
659  };
660
661  /**
662  * FUNCION PARA ELIMINAR EL EXTRACTOR
663  * SE ENVIA DATOS A LA BASE DE DATOS
664  * USUARIO
665  * NOMBRE DEL EXTRACTOR
666  * MUESTRA MENSAJES DE ACUERDO A CASOS
667  */
668  eliminarExtractor = () => { ...
778  };
779
```

Figura 2.35: Funciones del Componente Mis Extractores.

La función render del componente Servicio Mis Extractores es similar Servicio Crear Extractor Personalizado, pero la diferencia radica en que se agregan componentes internos para la visualización de todos los extractores del usuario y las características de un extractor en particular, como se puede apreciar en la Figura 2.36.

```

1780 render() {
1781   /**
1782    * CREACION DE COMPONENTES INTERNOS
1783    */
1784   let TodosMisExtractores = this._MisExtractores();
1785   let VerExtractorModal = this._modalVerExtractor();
1786
1787   let CrearNodoModal = this._modalCrearNodo();
1788   let GuardarNodoModal = this._modalGuardarNodo();
1789   let EliminarNodoModal = this._modalEliminarNodo();
1790   let VerNodoModal = this._modalVerNodo();
1791   let EliminarExtractorModal = this._modalEliminarExtractor();
1792
1793   // UTILIDADES
1794   let LOADER = this._Loader();
1795   let OpcionesDescarga = this._opcionesDescarga();
1796   let BotonItemsExtraidos = this._botonItemsExtraidos();
1797   let ModalItemsExtraidos = this._modalItemsExtraidos();
1798   let BotonItemsExtraidosExtractor = this._botonItemsExtraidosExtractor();
1799   let ModalItemsExtraidosExtractor = this._modalItemsExtraidosExtractor();
1800   let OpcionesDescargaExtractor = this._opcionesDescargaExtractor();
1801   return (
1802     <div>
1803       {LOADER}
1804       {OpcionesDescarga}
1805       {BotonItemsExtraidos}
1806       {ModalItemsExtraidos}
1807       {OpcionesDescargaExtractor}
1808       {BotonItemsExtraidosExtractor}
1809       {ModalItemsExtraidosExtractor}
1810       {CrearNodoModal}
1811       {GuardarNodoModal}
1812       {EliminarNodoModal}
1813       {VerNodoModal}
1814       {VerExtractorModal}
1815       {EliminarExtractorModal}
1816
1817       /* SECCION MIS EXTRACTORES
1818        * CONTIENE LOS EXTRACTORES RESCATADOS DESDE LA BD
1819        */
1820       <div className="first-wireframe-misExtractores">
1821         <h2 className="titulo-white">Mis Extractores</h2>
1822         <div className="caja-extractores">{TodosMisExtractores}</div>
1823       </div>
1824     </div>
1825   );
1826 }
1827 }

```

Figura 2.36: Función Render del Componente Mis Extractores.

Para poder importar el componente en otros archivos, se define su exportación como se puede apreciar en la Figura 2.37. En este caso el componente Extractor de Twitter esta conectado con Redux, por lo tanto es capaz de hacer peticiones para cambiar variables de la Store principal de la aplicación y a su vez puede capturar variables de la Store.

```
1829 /**
1830  * ESTE COMPONENTE RENDERISA DEPENDIENDO DE LA STORE
1831  * DEPENDE DE:
1832  * SI EXISTE O NO USUARIO
1833  * SI EXISTE O NO EXTRACTORES DEL USUARIO
1834  * SI EXISTE O NO RESULTADOS EXTRACCION POR EL NODO
1835  * SI EXISTE O NO RESULTADOS EXTRACCION POR EL EXTRACTOR
1836  */
1837 const mapStateToProps = state => {
1838   let { extraccion } = state.ExtraccionPersonalizada;
1839   let { solicitud } = state.Usuario;
1840   let { extractorExtraccion } = state.ExtraccionExtractor;
1841   return {
1842     Extractores: state.MisExtractores,
1843     extraccionPersonalizada: extraccion,
1844     extraccionExtractor: extractorExtraccion,
1845     Usuario: solicitud.usuario
1846   };
1847 };
1848
1849 /**
1850  * ESTE COMPONENTE PUEDE ENVIAR PETICIONES A LA STORE:
1851  * PUEDE ENVIAR:
1852  * PETICION CARGAR MIS EXTRACTORES
1853  * PETICION SERVICIO PROBAR NODO
1854  * PETICION SERVICIO PROBAR EXTRACTOR
1855  * BORRAR DATOS DE PROBAR NODO
1856  * BORRAR DATOS DE PROBAR EXTRACTOR
1857  * UN MENSAJE ESTATICO
1858  * UN NUEVO MENSAJE
1859  */
1860 function matchDispatchToProps(dispatch) {
1861   return bindActionCreators(
1862     {
1863       cargarMisExtractores: misExtractores,
1864       peticionExtraccionPersonalizada: peticionExtraccionPersonalizada,
1865       borrarDatosExtraccionPersonalizada: ExtraccionPersonalizada,
1866       peticionExtraccionExtractor: peticionExtraccionExtractor,
1867       borrarDatosExtraccionExtractor: extraccionExtractor,
1868       mostrarMensaje: mostrarMensaje,
1869       mensajesEstatico: mensajes
1870     },
1871     dispatch
1872   );
1873 }
1874
1875 export default withRouter(
1876   connect(mapStateToProps, matchDispatchToProps)(MisExtractores)
1877 );
1878
```

Figura 2.37: Export del Componente Mis Extractores.

Se construyó el Servidor en Node.js donde se desarrollaron rutas de acceso o end-points para las respuestas a las peticiones desde la aplicación, tanto para prestar los Servicios como también las conexiones necesarias con la base de datos, por un tema de visualización y número de líneas de código se muestra sólo la definición de los end-points desarrollados, como se puede apreciar en la Figura 2.38.

```
71
72  app.post("/teco-api/registrarse", function(req, res) { ...
119  });
120
121  app.post("/teco-api/ingresar", function(req, res) { ...
178  });
179
180  app.post("/teco-api/verificarDatos", function(req, res) { ...
223  });
224
225  app.post("/teco-api/cambiarContrasena", function(req, res) { ...
264  });
265
266  app.post("/teco-api/mis_extractores", function(req, res) { ...
283  });
284
285  app.post("/teco-api/servicios/personalizado/guardar_extractor", function( ...
288  ) { ...
374  });
375
376  app.post("/teco-api/servicios/personalizado/editar_extractor", function( ...
379  ) { ...
509  });
510
511  app.post("/teco-api/servicios/personalizado/eliminar_extractor", function( ...
514  ) { ...
580  });
581
582  app.post("/teco-api/servicios/twitter", function(req, res) { ...
701  });
702
703  app.post("/teco-api/servicios/personalizado/probar_nodo", function(req, res) { ...
799  });
800
801  app.post("/teco-api/servicios/personalizado/probar_extractor", function( ...
804  ) { ...
913  });
```

Figura 2.38: End-Points del Servidor de la Aplicación TECO.

2.4.2.4. Validación y Verificación

Para la realización de validación y verificación de las funciones codificadas, se trabajó con un usuario real con conocimiento del lenguaje HTML en contacto con la aplicación, se le hizo una breve descripción del contenido y servicios ofrecidos por parte de la aplicación, invitándolo a seguir una serie de test unitarios, en esta oportunidad se especificaron los test relacionados con el Servicio de Extracción desde Twitter, Servicio de Mis Extractores y Manejo de Cuentas de Usuario:

T15. Buscar Tweets:

En la Tabla 2.15, se detalla los datos ingresados para el test unitario de la función Buscar Tweets, efectuando el test con el servicio en actividad y sin actividad, los datos principales para este test es el ingreso de una cadena de texto para realizar la búsqueda de los tweets mediante el Servicio de Extractor desde Twitter.

Tabla 2.15: Test Unitario para función Buscar Tweets.

Variables	Valores	Salida	Estado
Prueba con Servicios Activados		Extracción exitosa, aparecen botones para	Proceso
Cadena de Texto	piñera	Visualizar o Descargar los tweets obtenidos	Exitoso
Prueba con Servicios No Activados		Mensaje de Advertencia:	Proceso
Cadena de Texto	piñera	Error por conexión con el Servicio	Fallido

T16. Descargar Tweets Extraídos:

En la Tabla 2.16, se detalla los datos ingresados para el test unitario de la función Descargar Tweets Extraídos, este test se puede realizar luego de haber ejecutado exitosamente el Servicio de Extractor desde Twitter y dar click al botón que hace referencia a elegir el formato de descarga de los datos extraídos, los datos principales para este test son los tweets extraídos exitosamente desde el Servicio de Extractor desde Twitter.

Tabla 2.16: Test Unitario para función Descargar Datos Extraídos por Extractores.

Variables	Valores	Salida	Estado
Opción Archivo de Texto		Genera el archivo de Texto con los tweets clasificándolos por sus elementos.	Proceso
Tweets	tweets extraídos		Exitoso
Opción Archivo CSV		Genera el archivo de CSV con los tweets clasificándolos por sus elementos.	Proceso
Tweets	tweets extraídos		Exitoso

T17. Visualizar Tweets Extraídos:

En la Tabla 2.17, se detalla los datos ingresados para el test unitario de la función Visualizar Tweets Extraídos, este test se puede realizar luego de haber ejecutado exitosamente el Servicio de Extractor desde Twitter y dar click al botón que hace referencia a visualizar los datos extraídos, los datos principales para este test son los tweets extraídos exitosamente desde el Servicio de Extractor desde Twitter.

Tabla 2.17: Test Unitario para función Visualizar Tweets Extraídos.

Variables	Valores	Salida	Estado
Tweets	tweets extraídos	Muestra modal con los tweets extraídos clasificados por los componentes de cada tweet.	Proceso Exitoso

T18. Cargar Mis Extractores:

En la Tabla 2.18, se detalla los datos ingresados para el test unitario de la función Cargar Mis Extractores, efectuando el test con un usuario sin y con extractores creados y guardados, los datos principales para este test es el ingreso de un usuario haciendo referencia a la opción del menú Mis Extractores presentes para los usuarios que han ingresado exitosamente a la plataforma.

Tabla 2.18: Test Unitario para función Cargar Mis Extractores.

Variables	Valores	Salida	Estado
Usuario con Extractores		Muestra listado de extractores rescatados desde la base de datos.	Proceso Exitoso
Usuario	usuarioprueba		
Usuario sin Extractores		Muestra texto invitando a crear extractores con el servicio de crear extractor personalizado.	Proceso Exitoso
Usuario	usuarioprueba		

T19. Eliminar Extractor Personalizado:

En la Tabla 2.19, se detalla los datos ingresados para el test unitario de la función Eliminar Extractor Personalizado, este test se puede realizar luego de haber creado y guardado al menos 1 extractor personalizado con el Servicio Crear Extractor Personalizado, posteriormente se da click sobre el extractor que se desea eliminar para visualizar sus características, terminando con dar click a la opción Eliminar Extractor, los datos principales para este test es el nombre del extractor que se desea eliminar.

Tabla 2.19: Test Unitario para función Eliminar Extractor Personalizado.

Variables	Valores	Salida	Estado
Usuario	usuarioprueba	Eliminación del extractor exitosa	Proceso Exitoso
Nombre del Extractor	La Tercera		

T20. Editar Extractor Personalizado:

En la Tabla 2.20, se detalla los datos ingresados para el test unitario de la función Editar Extractor Personalizado, este test se puede realizar luego de haber creado y guardado al menos 1 extractor personalizado con el Servicio Crear Extractor Personalizado, posteriormente se da click sobre el extractor que se desea modificar para visualizar sus características, terminando con dar click a la opción Guardar Extractor, los datos principales para este test son los datos modificados de un extractor creado recientemente.

Tabla 2.20: Test Unitario para función Editar Extractor Personalizado.

Variables	Valores	Salida	Estado
Usuario	usuarioprueba	Guardado de modificaciones del extractor exitoso	Proceso Exitoso
Nombre del Extractor	La Tercera 2		
Descripción	Extrae noticias ultimo minuto		
Url	http://www.latercera.com		
Nodos Extractores	Noticias Ultimo Minuto		
Mis Extractores	La Tercera		
Usuario	usuarioprueba	Mensaje de Advertencia: El nombre del extractor ya existe	Proceso Fallido
Nombre del Extractor	La Tercera		
Descripción	Extrae noticias ultimo minuto		
Url	http://www.latercera.com		
Nodos Extractores	Noticias Ultimo Minuto		
Mis Extractores	La Tercera		
Usuario	usuarioprueba	Mensaje de Advertencia: El extractor debe contener al menos 1 nodo extractor.	Proceso Fallido
Nombre del Extractor	La Tercera		
Descripción	Extrae noticias ultimo minuto		
Url	http://www.latercera.com		
Nodos Extractores	sin nodos		
Mis Extractores	La Tercera		

T21. Registrar Usuario:

En la Tabla 2.21, se detalla los datos ingresados para el test unitario de la función Registrar Usuario, efectuando el test con los datos necesarios para registrar un nuevo usuario a la plataforma mediante la opción del menú Registro, los datos principales para este test es el ingreso de los datos de un nuevo usuario para el registro en la plataforma.

Tabla 2.21: Test Unitario para función Registrar Usuario.

Variables	Valores	Salida	Estado
Correo Electrónico	usuarioprueba@gmail.com	Registro de nuevo usuario exitoso	Proceso Exitoso
Usuario	usuarioprueba		
Contraseña	usuarioprueba		
Primer Nombre	Super		
Segundo Nombre	Usuario		
Primer Apellido	Tests		
Segundo Apellido	Pruebas		
Pista Palabra Clave	superusuario		
Palabra Clave	superusuario		
Correo Electrónico	usuarioprueba@gmail.com	Mensaje de Advertencia: El Usuario o el Correo Electrónico ya existen	Proceso Fallido
Usuario	usuarioprueba2		
Contraseña	usuarioprueba		
Primer Nombre	Super2		
Segundo Nombre	Usuario2		
Primer Apellido	Tests2		
Segundo Apellido	Pruebas2		
Pista Palabra Clave	superusuario2		
Palabra Clave	superusuario2		
Correo Electrónico	usuarioprueba2@gmail.com	Mensaje de Advertencia: El Usuario o el Correo Electrónico ya existen	Proceso Fallido
Usuario	usuarioprueba		
Contraseña	usuarioprueba		
Primer Nombre	Super2		
Segundo Nombre	Usuario2		
Primer Apellido	Tests2		
Segundo Apellido	Pruebas2		
Pista Palabra Clave	superusuario2		
Palabra Clave	superusuario2		

T22. Ingresar Usuario:

En la Tabla 2.22, se detalla los datos ingresados para el test unitario de la función Ingresar Usuario, efectuando el test con los datos necesarios para ingresar como usuario a la plataforma mediante la opción del menú Ingresar, los datos principales para este test es el ingreso del usuario y contraseña.

Tabla 2.22: Test Unitario para función Ingresar Usuario.

Variables	Valores	Salida	Estado
Usuario	usuarioprueba	Ingreso a la plataforma exitoso	Proceso
Contraseña	usuarioprueba	Bienvenido Super Usuario Tests Pruebas	Exitoso
Usuario	usuarioprueba2	Mensaje de Advertencia:	Proceso
Contraseña	usuarioprueba	Usuario o Contraseña Incorrectas	Fallido
Usuario	usuarioprueba	Mensaje de Advertencia:	Proceso
Contraseña	usuarioprueba25	Usuario o Contraseña Incorrectas	Fallido

T23. Cambiar Contraseña:

En la Tabla 2.23, se detalla los datos ingresados para el test unitario de la función Cambiar Contraseña, efectuando el test con datos específicos relacionados a un usuario mediante la opción Cambiar Contraseña presente en la pestaña del menú Ingresar, los datos principales para este test es el ingreso del usuario, correo electrónico, pista para la palabra clave, la palabra clave y posteriormente el ingreso de una nueva contraseña.

Tabla 2.23: Test Unitario para función Cambiar Contraseña.

Variables	Valores	Salida	Estado
Correo Electrónico	usuarioprueba@gmail.com	Cambio de Contraseña Exitoso	Proceso Exitoso
Usuario	usuarioprueba		
Pista Palabra Clave	superusuario		
Palabra Clave	superusuario		
Nueva Contraseña	superusuario		
Correo Electrónico	usuarioprueba@gmail.com	Mensaje de Advertencia: Usuario o Correo Electrónico Incorrectos	Proceso Fallido
Usuario	usuarioprueba25		
Pista Palabra Clave	superusuario		
Palabra Clave	superusuario		
Nueva Contraseña	superusuario		
Correo Electrónico	usuarioprueba@gmail.com	Mensaje de Advertencia: Palabra Clave Incorrecta	Proceso Fallido
Usuario	usuarioprueba		
Pista Palabra Clave	superusuario		
Palabra Clave	superusuario45		
Nueva Contraseña	superusuario		

Capítulo 3

Conclusiones

El objetivo general de este proyecto era desarrollar una aplicación extractora de textos desde páginas web y APIs públicas basada en criterios de búsqueda. Para esto, fue necesario cumplir con los objetivos específicos impuestos, los que podríamos separar en cuatro grupos: Estudio de conceptos generales, Estudio sobre herramientas con propósitos similares, Estudio de herramientas de programación y Desarrollo de la aplicación Text Extraction Community.

Estudio de conceptos generales

Los estudios realizados en esta etapa fueron necesarios para contextualizar el entorno en que puede ser utilizada la aplicación, además, fue necesario indagar sobre las estrategias utilizadas para la extracción de textos desde las páginas web para concluir que la estrategia de extracción Scrapping requería un grado de complejidad media en comprensión y tiempo de implementación, ajustándose al conocimiento del desarrollador y los tiempos estipulados en la Carta Gantt, por lo tanto fue elegida como base para una solución a este proyecto.

Estudio sobre herramientas con propósitos similares

Dado los estudios realizados sobre las herramientas existentes, predominaron las que ofrecen variadas funcionalidades a cambio de una suscripción pagada, las cuales sobrepasan el conocimiento del desarrollador y los tiempos estipulados para el proyecto, por esta razón se eligió la herramienta Scraper Google Extension como base para el diseño y el desarrollo de la aplicación, ya que sus funcionalidades reflejan el concepto de Scrapping con un grado de complejidad media en comprensión.

Estudio de herramientas de programación

Para la realización del proyecto se estudiaron como herramientas de programación React.js, Node.js y MongoDB, con el propósito de acotar la necesidad de aprender nuevos lenguajes de programación, optando por el lenguaje Javascript, propiedad en común entre estas tres herramientas, como base fundamental para el desarrollo de la aplicación Text Extraction Community, permitiendo agilizar el proceso de creación e implementación de las funcionalidades requeridas en este proyecto, complementando el proceso con los módulos para libre uso ofrecidos por la Node Package Manager y la documentación almacenada en la plataforma de GitHub.

Desarrollo de la aplicación Text Extraction Community

En esta etapa se comenzó haciendo el primer incremento del proceso de desarrollo de la aplicación, el cual correspondía al análisis, el diseño, la codificación, validación y verificación del Servicio Crear un Extractor Personalizado valorado con una complejidad alta, para terminar con la realización del segundo incremento correspondiente al mismo procedimiento anterior orientándolo hacia el Servicio Mis Extractores y Servicio Extractor desde Twitter valorados con una complejidad media y baja respectivamente, concretando la aplicación Text Extraction Community en un 100 % respecto a los requerimientos funcionales y no funcionales planteados en este proyecto.

Trabajos futuros

Para los trabajos futuros se podría realizar el desarrollo de una API basada en los servicios y esquematización de los datos presentes en la aplicación, quedando a disposición del público en general, desarrollar funcionalidades para compartir extractores, generar un sistema de concurrencia de extracciones y un sistema para extracciones programadas.

Referencias

- ChromeWebStore (2017). Chrome web store - scraper google extension.
<https://chrome.google.com/webstore/detail/scraper/mbigbapnjcgaffohmbkdlecaccepngjd>.
- Ekman, P. (1993). Facial expression and emotion. *American psychologist*, 48(4):384.
- GitHub (2017). Github official website. <https://github.com/>.
- Github-Axios (2017). axios http promise requests.
<https://github.com/axios/axios>.
- Github-Bcrypt (2017). Bcrypt hash passwords.
<https://github.com/kelektiv/node.bcrypt.js/>.
- Github-Body-Parser (2017). Body parsing middleware.
<https://github.com/expressjs/body-parser>.
- Github-Connect-Mongo (2017). Mongoddb session store.
<https://github.com/jdesboeufs/connect-mongo>.
- Github-Cors (2017). Cors requests. <https://github.com/expressjs/cors>.
- Github-Create-React-App (2017). Create react app.
<https://github.com/facebook/create-react-app>.
- Github-Express (2017). Express middleware.
<https://github.com/expressjs/express>.
- Github-Mongoose (2017). Mongoose middleware.
<https://github.com/Automattic/mongoose>.

Github-Nitghtmare (2017). Nightmare middleware.
<https://github.com/segmentio/nightmare>.

Github-React (2017). react javascript library user interfaces.
<https://github.com/facebook/react>.

Github-React-Bootstrap (2017). React components create by bootstrap.
<https://github.com/react-bootstrap/react-bootstrap>.

Github-React-CSV (2017). React csv document.
<https://github.com/abdenmour/react-csv>.

Github-React-DOM (2017). React dom rendering paths.
<https://github.com/facebook/react/tree/master/packages/react-dom>.

Github-React-Redux (2017). React redux bindings.
<https://github.com/reactjs/react-redux>.

Github-React-Router (2017). React declarative routing.
<https://github.com/ReactTraining/react-router>.

Github-React-Router-DOM (2017). Dom bindings for react router.
<https://github.com/ReactTraining/react-router/tree/master/packages/react-router-dom>.

Github-Redux (2017). Redux state container. <https://github.com/reactjs/redux>.

Github-Redux-Logger (2017). Logger middleware.
<https://github.com/evgenyrodionov/redux-logger>.

Github-Redux-Thunk (2017). Thunk middleware.
<https://github.com/gaearon/redux-thunk>.

Github-Semantic-UI-React (2017). Semantic ui components.
<https://github.com/Semantic-Org/Semantic-UI-React>.

Github-Yarn (2017). Yarn dependency management.
<https://github.com/yarnpkg/yarn>.

- González, J. A. M. (2004). El contenido de los documentos textuales: su análisis y representación mediante el lenguaje natural. *Rev. Esp. Doc. Cient*, 27(4):573.
- GoogleSearch (2017). Googlesearch website. <https://www.google.com/search/howsearchworks/>.
- Import.io (2017). Import.io. <https://www.import.io>.
- Instalación-MongoDB (2017). Instalación mongodb. <https://victorroblesweb.es/2016/11/26/instalar-mongodb-windows/>.
- Jiménez Zafra, S. M., Martínez Cámara, E., Martín Valdivia, M. T., and Molina González, M. D. (2015). Tratamiento de la negación en el análisis de opiniones en español. *Procesamiento del Lenguaje Natural*, (54).
- Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.
- Martínez, C., Elgueta, J., Segura, A., Vidal-Castro, C., and Rubio, C. (2015). What do our children read about? affect analysis of chilean school texts. *Int. J. Comput. Linguistics Appl.*, 6(1):63–79.
- Microsoft (2017). Diagrama actividades microsoft. <https://msdn.microsoft.com/es-es/library/dd409360.aspx>.
- MongoDB (2017). Mongodb official website. <https://www.mongodb.com/es>.
- MongoDBArticles (2017). Mongodb official website. <https://www.mongodb.com/white-papers>.
- NodeCollection (2017). Node.js collection. <https://medium.com/the-node-js-collection>.
- NodeJS (2017). Nodejs official website. <https://nodejs.org/es/>.
- Octoparse (2017). Octoparse. <https://www.octoparse.com>.
- ParseHub (2017). Parsehub. <https://www.parsehub.com>.

- Plutchik, R. (1980). A general psychoevolutionary theory of emotion. *Theories of emotion*, 1(3-31):4.
- Precios-Import.io (2017). Precios import.io. <https://www.import.io/standard-plans/>.
- Precios-Octoparse (2017). Precios octoparse. <https://www.octoparse.com/Pricing>.
- Precios-ParseHub (2017). Precios parsehub. <https://www.parsehub.com/pricing>.
- ReactJS (2017). Reactjs official website. <https://reactjs.org/>.
- ReactJS-Articles (2017). Reactjs official website. <https://reactjs.org/community/articles.html>.
- Scrapergoogleextension (2017). Scrapper google extension. <https://chrome.google.com/webstore/detail/scraper/mbigbapnjcgaffohmbkdlecacepngjd>.
- Sommerville, I. (2011). *Ingeniería de Software*. Pearson educación, México, novena edition.
- TechopediaDefinition-WebCrawler (2017). Techopediade finition-webcrawler. <https://www.techopedia.com/definition/10008/web-crawler>.
- TechopediaDefinition-WebScraping (2017). Techopediade finition-webscraping. <https://www.techopedia.com/definition/5212/web-scraping>.
- Twitter-Search (2017). Twitter search. <https://twitter.com/search-home>.
- Vásquez, A. C., Quispe, J. P., Huayna, A. M., et al. (2009). Procesamiento de lenguaje natural. *Revista de investigación de Sistemas e Informática*, 6(2):45–54.
- VisualStudioCode (2017). Microsoft visual studio code. <https://code.visualstudio.com/>.

Apéndices

A. Manual de Usuario

A.1. Proceso de Instalación

Instalar MongoDB: (Instalación-MongoDB, 2017)

- Descargar MongoDB (<https://www.mongodb.com/>)
- Crear el directorio C:\mongodb\data
- Crear el directorio C:\mongodb\data\db
- Crear el directorio C:\mongodb\log
- Abrir terminal como administrador
- Dirigirse a donde instalaste MongoDB \bin
- Ejecutar MongoDB, Figura 3.1.
- Inicio de MongoDB, Figura 3.2.

```
C:\> Administrador: Símbolo del sistema
C:\mongodb\bin>mongod --dbpath "C:\mongodb\data\db" --logpath "C:\mongodb\log\mongo.log"
```

Figura 3.1: Ejecutar MongoDB

```
2016-11-04T13:42:50.647+0100 I CONTROL [initandlisten] db version v3.2.10
2016-11-04T13:42:50.647+0100 I CONTROL [initandlisten] git version: 79d9b3ab5ce
20f51c272b4411202710a002d0317
2016-11-04T13:42:50.647+0100 I CONTROL [initandlisten] OpenSSL version: OpenSSL
1.0.1t-fips 3 May 2016
2016-11-04T13:42:50.647+0100 I CONTROL [initandlisten] allocator: tcmalloc
2016-11-04T13:42:50.648+0100 I CONTROL [initandlisten] modules: none
2016-11-04T13:42:50.648+0100 I CONTROL [initandlisten] build environment:
2016-11-04T13:42:50.648+0100 I CONTROL [initandlisten] distmod: 2008plus-ss
l
2016-11-04T13:42:50.648+0100 I CONTROL [initandlisten] distarch: x86_64
2016-11-04T13:42:50.648+0100 I CONTROL [initandlisten] target_arch: x86_64
2016-11-04T13:42:50.648+0100 I CONTROL [initandlisten] options: {}
2016-11-04T13:42:50.667+0100 I STORAGE [initandlisten] wiredtiger_open config:
create,cache_size=4G,session_max=20000,eviction=(threads_max=4),config_base=false,
statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snapp
y),file_manager=(close_idle_time=100000),checkpoint=(wait=60,log_size=2GB),stati
stics_log=(wait=0),
2016-11-04T13:42:51.516+0100 I NETWORK [HostnameCanonicalizationWorker] Startin
g hostname canonicalization worker
2016-11-04T13:42:51.516+0100 I FTDC [initandlisten] Initializing full-time d
iagnostic data capture with directory 'C:/data/db/diagnostic.data'
2016-11-04T13:42:51.665+0100 I NETWORK [initandlisten] waiting for connections
on port 27017
```

Figura 3.2: Inicio de MongoDB en puerto 27017 por default

Instalar NodeJS:

- Descargar NodeJS (<https://nodejs.org/es/>)
- Completar setup de instalación
- Abrir consola de comandos
- Instalar Yarn , Figura 3.3.

```
npm install --global yarn
```

Figura 3.3: Instalar Yarn vía Node Package Manager

Iniciar Aplicación:

- Descargar y Descomprimir TECOapp.rar, Figura 3.4.

Nombre	Fecha de modifica...	Tipo
📁 Cliente	30-09-2017 22:04	Carpeta de archivos
📁 Servidor	12-03-2018 18:38	Carpeta de archivos
📄 ReadMe	13-03-2018 3:06	Documento de tex...

Figura 3.4: Carpetas de la Aplicación TECO

- Abrir terminal de comando en la carpeta Cliente e Inicialo, Figura 3.5.

```
\TECOapp\Cliente> yarn start
```

Figura 3.5: Iniciar Interfaz Web de la Aplicación TECO

- Abrir terminal de comando en la carpeta Servidor e Inicialo, Figura 3.6.

```
\TECOapp\Servidor> yarn dev
```

Figura 3.6: Iniciar Servidor de la Aplicación TECO

- Ya puedes ingresar a la página local de la aplicación <http://localhost:3000>

A.2. Interfaces de la Aplicación TECO

Páginas de Inicio:

Las páginas de inicio muestran un menú de navegación para que el usuario interacciones con los demás componentes, también muestran información resumida de los Servicios que ofrece la aplicación, por ultimo describe los formatos de descarga al usar los Servicios, como se puede apreciar en las Figuras 3.7, 3.8, 3.9 y 3.10.

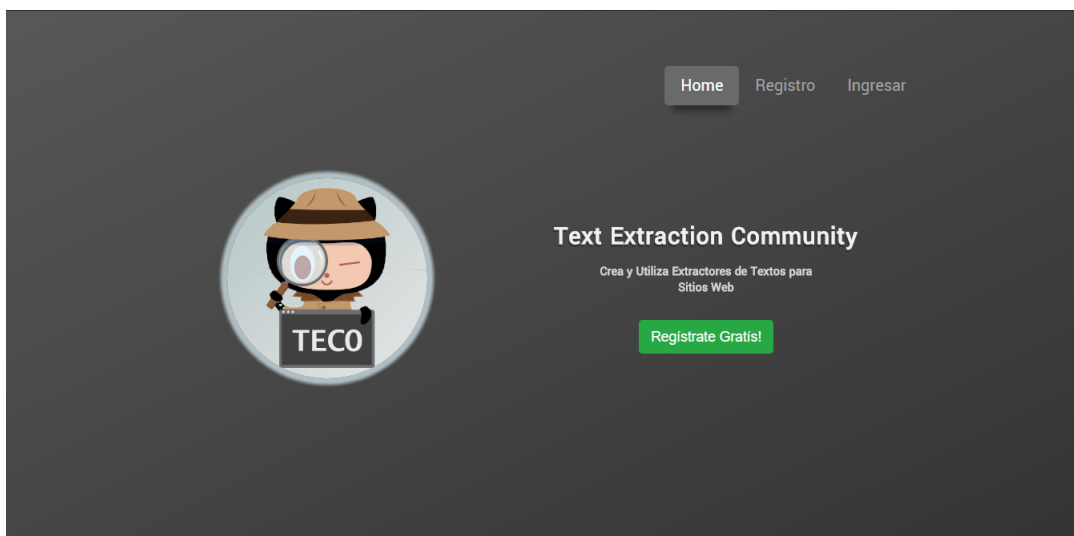


Figura 3.7: Página inicio de la aplicación TECO.



Figura 3.8: Información reducida del extractor de twitter de la página de inicio.

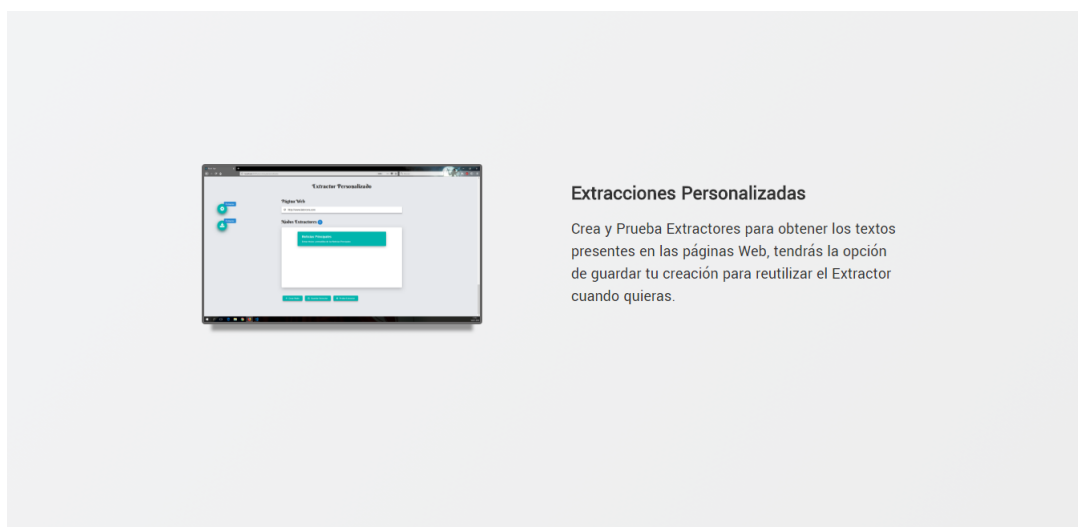


Figura 3.9: Información reducida del extractor personalizado de la página de inicio.



Figura 3.10: Página de información formatos de descargas.

Registro de nuevo usuario en la Plataforma:

Página para registrar un nuevo usuario en la plataforma, para realizar la petición se pide ingresar: Usuario, Contraseña, Primer Nombre, Segundo Nombre, Primer Apellido, Segundo Apellido, Pista para la Palabra Clave, Palabra Clave y un Correo Electrónico, para luego enviar la petición a través del botón Completar Registro ubicado al final del formulario de registro, como se puede apreciar en la Figura 3.11.



The image shows a user registration form on a dark-themed website. At the top right, there are navigation links for 'Home', 'Registro' (highlighted), and 'Ingresar'. The main heading is 'Registro de Usuario'. The form includes the following fields:

- Usuario**: A text input field with a user icon on the left.
- Contraseña**: A text input field with a magnifying glass icon on the left.
- Primer Nombre**: A text input field.
- Segundo Nombre**: A text input field.
- Primer Apellido**: A text input field.
- Segundo Apellido**: A text input field.

Figura 3.11: Página registro de nuevo usuario a la plataforma.

Ingreso de Usuario a la Plataforma:

Página para ingresar Usuario y Contraseña enviando la información a través del botón Ingresar, en el caso de tener problemas con la contraseña, existe la opción de cambiar la contraseña, como se puede apreciar en la Figura 3.12.

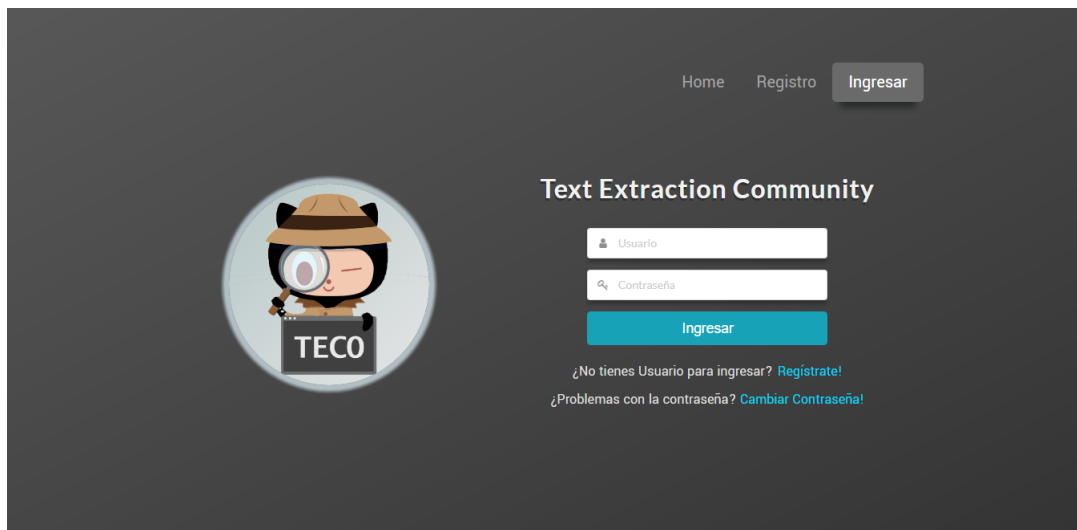


Figura 3.12: Página ingreso de usuario a la plataforma.

Proceso Cambio de Contraseña:

Para el cambio de contraseña se debe cumplir una serie de pasos, primero se pide ingresar el Usuario y Correo Electrónico enviándolos mediante datos personales como se puede apreciar en la Figura 3.13. En el caso de que los datos personales sean correctos, se pide ingresar la Palabra Clave registrada por el usuario, con fin de ayudar al usuario se muestra una Pista para ingresar correctamente la Palabra clave como se puede apreciar en la Figura 3.14. En el caso de ingresar la Palabra Clave correctamente se procede al último paso que consiste en ingresar una Nueva Contraseña para poder ingresar con el Usuario, como se puede apreciar en la Figura 3.15.

Home Registro Ingresar

Ingreso Datos Personales

Usuario

Correo Electrónico

Enviar Datos

[← Volver a Ingreso de Usuario](#)

Figura 3.13: Página ingreso de datos personales.

Home Registro Ingresar

Ingreso Palabra Clave

Pista para la Palabra Clave :
Super Usuario

Palabra Clave

Verificar Palabra Clave

[← Volver a Ingreso de Usuario](#)

Figura 3.14: Página ingreso de palabra clave.

Home Registro Ingresar

Ingreso Nueva Contraseña

Nueva Contraseña

Cambiar Contraseña

[← Volver a Ingreso de Usuario](#)

Figura 3.15: Página ingreso de nueva contraseña.

Ingreso Exitoso a la Plataforma:

En un ingreso exitoso a la plataforma, el menú de navegación cambia otorgando acceso a los Servicios al usuario, aparecen la opción Servicios y Mis Extractores en el menú de navegación, como se puede apreciar en la Figura 3.16.

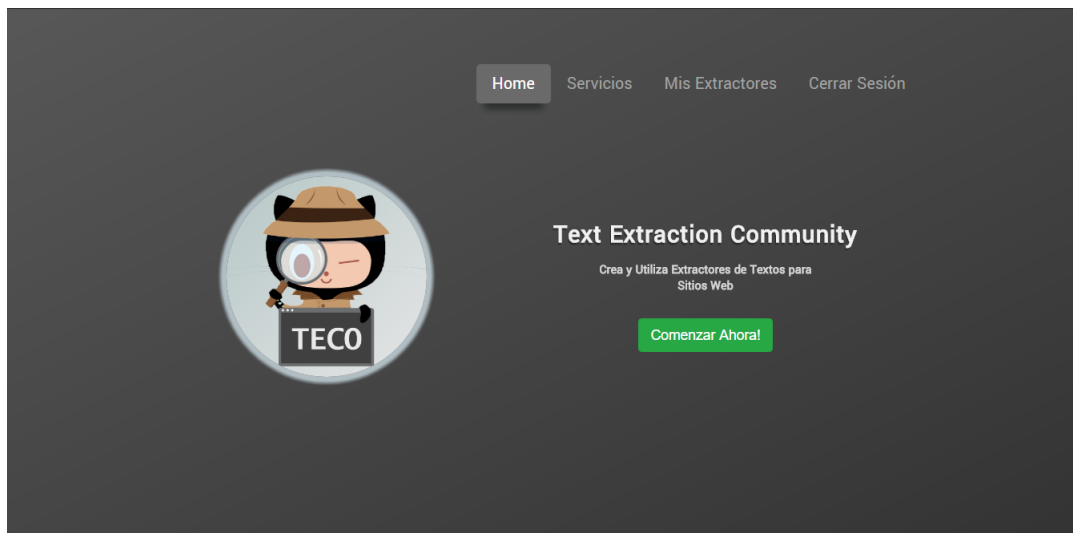


Figura 3.16: Página inicio usuario con ingreso exitoso en la plataforma.

Servicios de la Plataforma:

Página para que el Usuario seleccione entre las opciones Extractor de Twitter y Crear Extractor Personalizado como se puede apreciar en la Figura 3.17.



Figura 3.17: Página de selección de servicio.

Servicio Extractor de Twitter:

Al seleccionar el Servicio de Extractor de Twitter aparece una página con información de cómo usar el extractor representada en la Figura 3.18. Luego se describen los operadores que pueden ser utilizados para que la búsqueda de los tweets sea más específica, como se puede apreciar en la Figura 3.19. Luego aparece la página donde se encuentra el extractor donde se pide ingresar una cadena de texto para empezar la búsqueda de los tweets con la tecla ENTREAR, como se puede apreciar en la Figura 3.20. Posteriormente se aprecia el uso del extractor obteniendo resultados dando las opciones al usuario de visualizar o descargar los tweets obtenidos representadas en las Figuras 3.21, 3.22, 3.23 respectivamente.

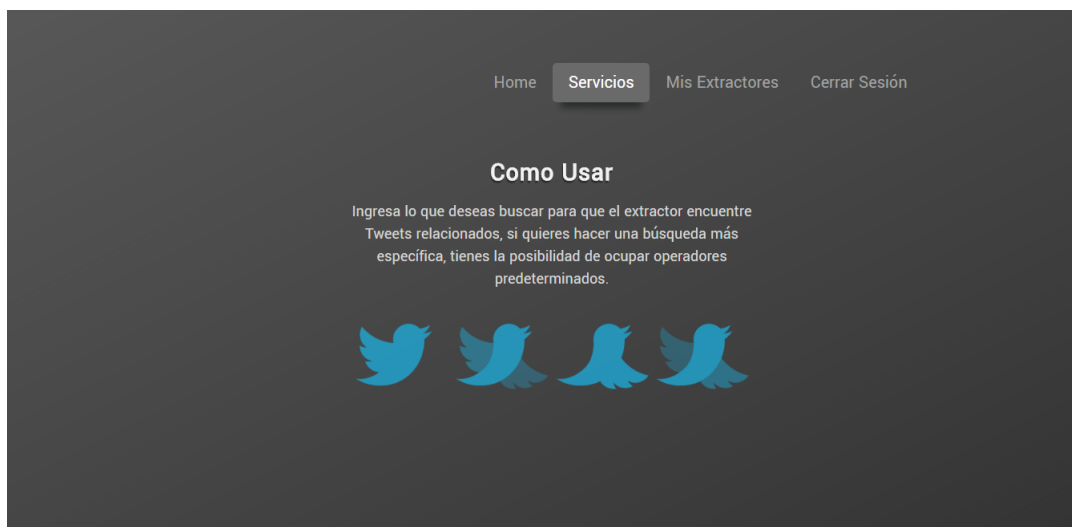


Figura 3.18: Página información cómo usar extractor twitter.

Operadores de Búsqueda

Operador	Encuentra Tweets ...
Texto normal	que contengan el texto que ingresaste en cualquier parte del tweet.
"hora feliz"	que contengan la frase exacta "hora feliz".
amor OR odio	que contengan ya sea "amor" u "odio" (o ambos).
cerveza -raiz	que contengan "cerveza" pero no "raiz".
#2018	conteniendo el hashtag "2018".
from:ucscconcepcion	enviados por "UCSC Concepción",tienes que asegurarte de ingresar el nombre del usuario, por ejemplo @sebastianpinera.
to:ucscconcepcion	enviados a "UCSC Concepción",tienes que asegurarte de ingresar el nombre del usuario, por ejemplo @sebastianpinera.
@ucscconcepcion	que hagan referencia a "ucscconcepcion".
"hora feliz" near:concepcion	que contengan la frase exacta "hora feliz" y sean enviados cerca de "concepcion".
near:concepcion within:15mi	enviados a 15 millas de "concepcion".

Figura 3.19: Página operadores de búsqueda para extractor de twitter.

Extractor de Twitter



Extrae de cada Tweet :

Usuario, Nombre del Usuario, Mensaje,
 Fecha, Cantidad de Comentarios,
 Retweets y de Favoritos.

Figura 3.20: Página del extractor de twitter.



Figura 3.21: Página de ejemplo del extractor de twitter.



Figura 3.22: Página de visualización de tweets obtenidos.



Figura 3.23: Página de selección de formato de descarga de tweets obtenidos.

Servicio Crear Extractor Personalizado:

Al seleccionar el Servicio de Crear Extractor Personalizado aparece una página con información de cómo usar el extractor representada en la Figura 3.24. Luego se describe la terminología en la que esta basada el proceso de creación del extractor, como se puede apreciar en la Figura 3.25. Luego aparece la página donde se encuentran recomendaciones para aumentar la probabilidad de una creación de un extractor que extraiga los textos que el usuario desea, como se puede apreciar en la Figura 3.26. Posteriormente se aprecia la página del extractor donde se pide ingresar la Url de la página a la cual se desea extraer los textos para luego empezar a crear nodos extractores, luego de crear nodos se pueden probar y guardar, como también se puede probar y guardar el extractor, dando la opción al usuario de visualizar y descargar los resultados obtenidos al probar nodos o el extractor completo, representadas en las Figuras 3.27, 3.28, 3.29, 3.30, 3.31 y 3.32 respectivamente.

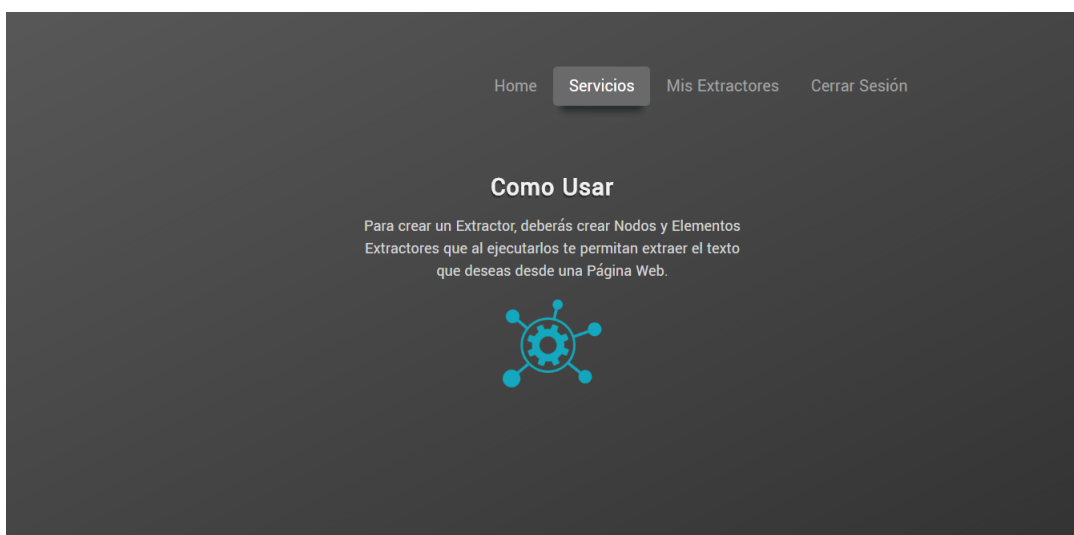


Figura 3.24: Página información cómo usar extractor personalizado.

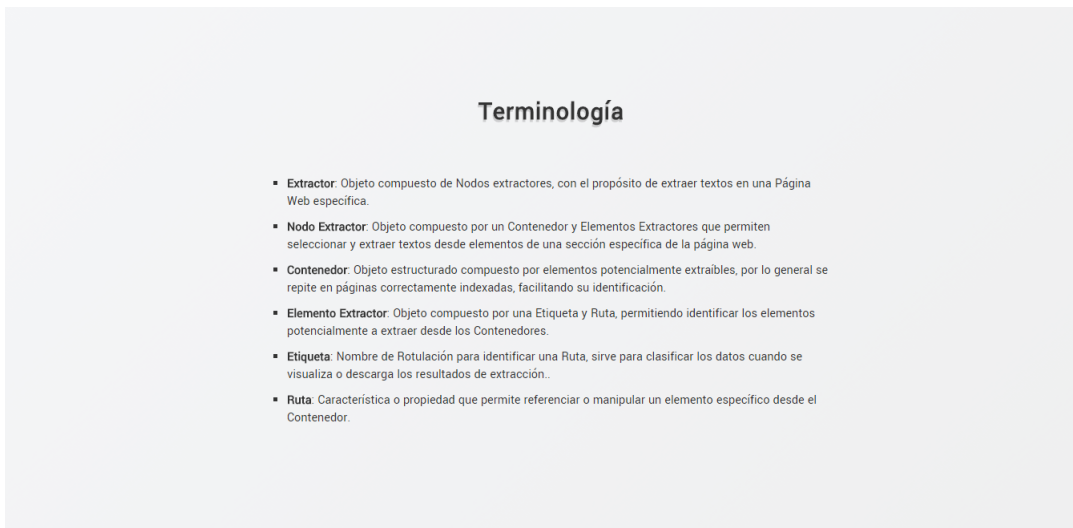


Figura 3.25: Página terminología en la que se basa el extractor personalizado.

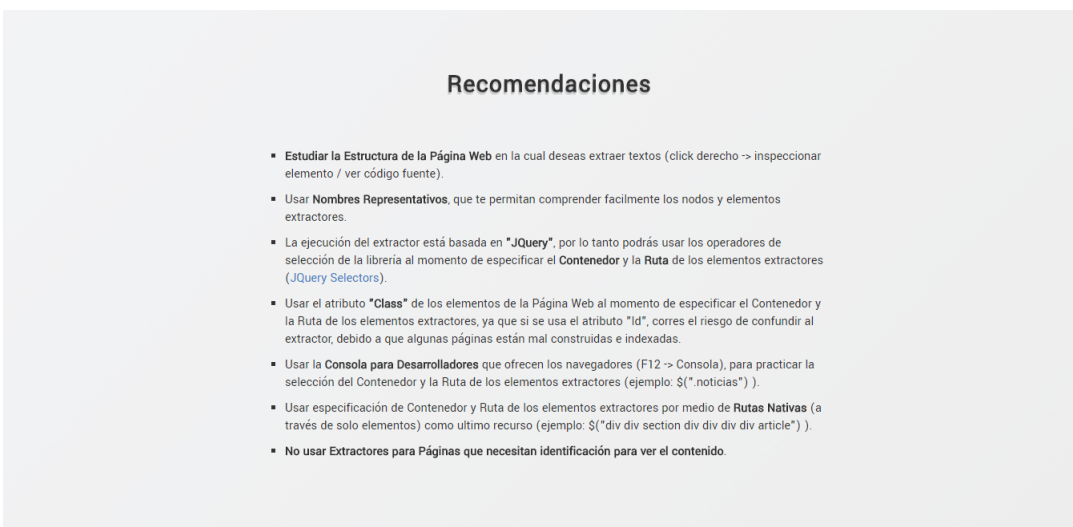


Figura 3.26: Página de recomendaciones para el usuario.

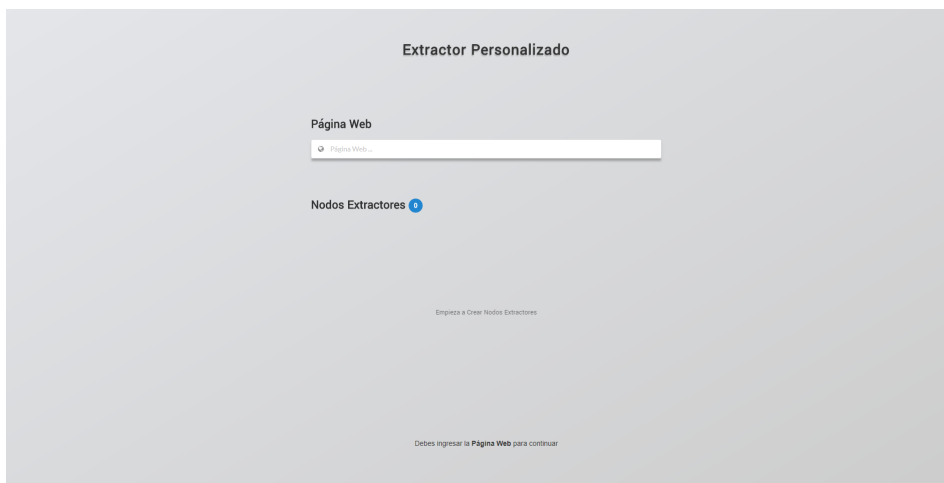


Figura 3.27: Página de creación del extractor personalizado.

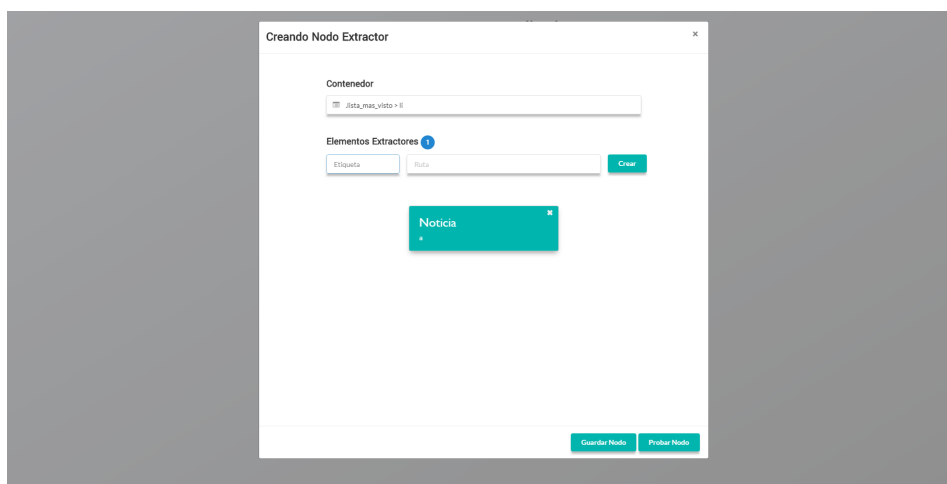


Figura 3.28: Página para creación de nodos extractores.

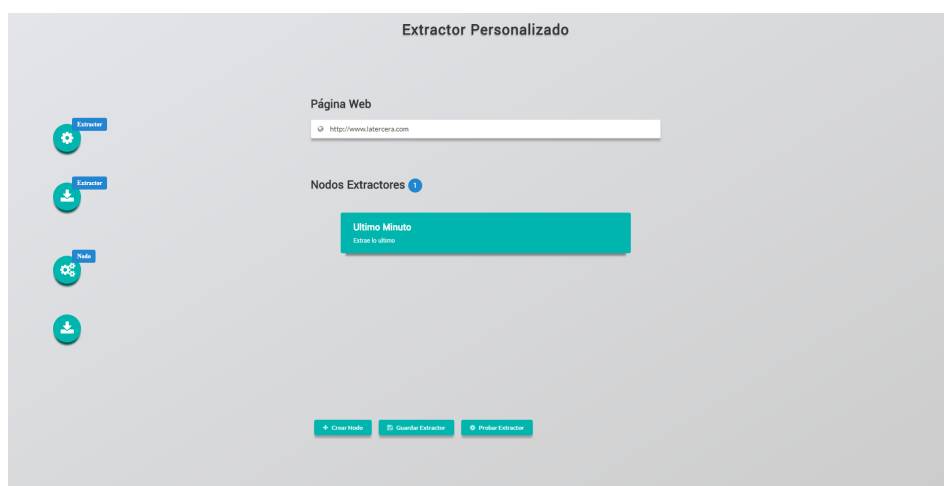


Figura 3.29: Página de ejemplo de creación y prueba de extractor.

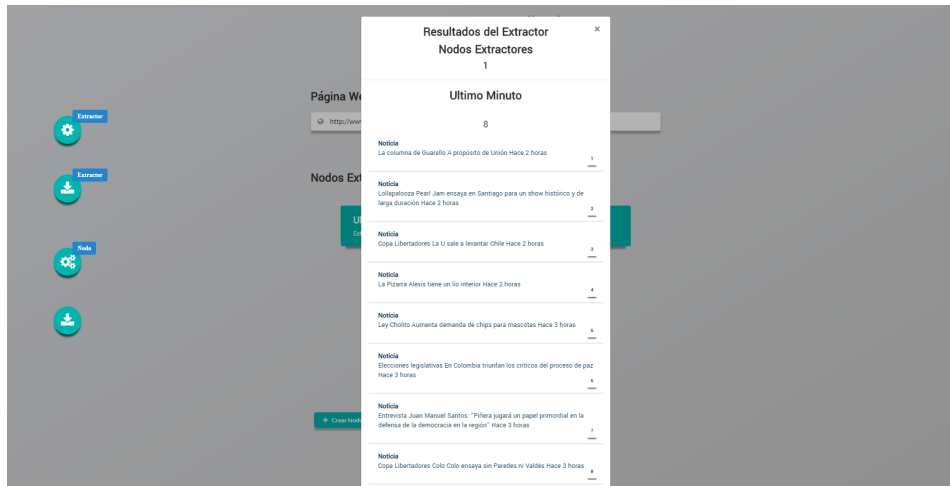


Figura 3.30: Página de visualización para los textos extraídos al probar el extractor.

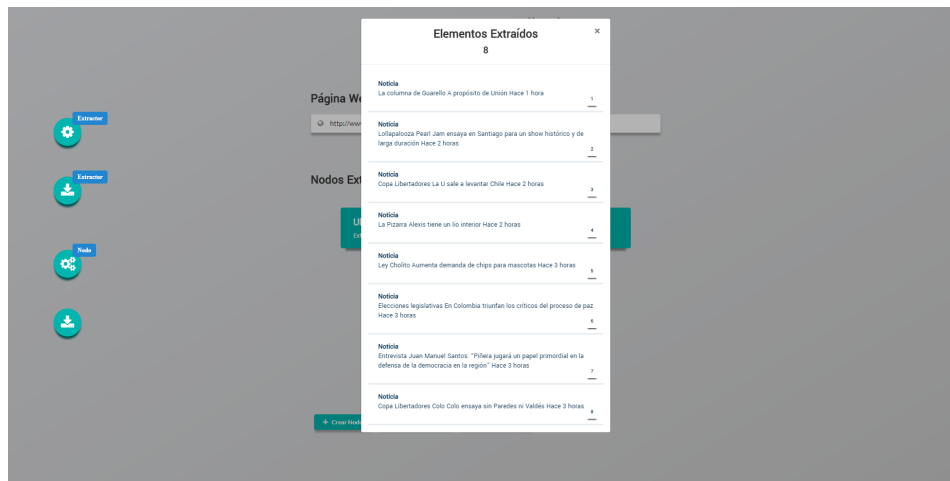


Figura 3.31: Página de visualización para los textos extraídos al probar nodos.

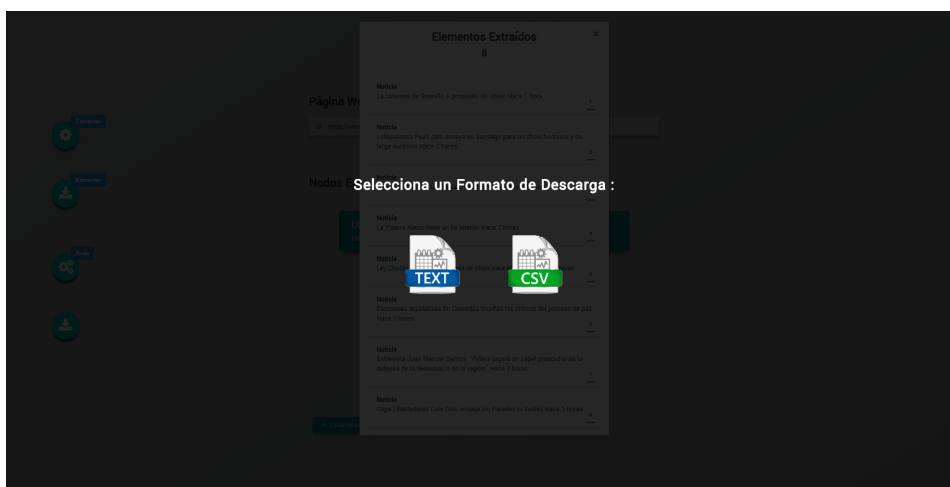


Figura 3.32: Página para selección de formato de descargas.

Servicio Mis Extractores:

Al seleccionar Mis Extractores en el menú de navegación se mostrará la página representada en la Figura 3.33. Donde se listan todos los extractores del usuario almacenados en la base de datos, para luego inspeccionar las características y registros que poseen cada extractor, dando la posibilidad de modificar, probar y eliminar el extractor como también los nodos y elementos extractores que lo componen como se puede apreciar en la Figura 3.34, este servicio es bastante similar al Servicio de Crear un Extractor Personalizado pero se parte de un nivel superior como lo es el extractor completo, es por esta razón que las páginas para crear, modificar nodos y elementos extractores son similares.

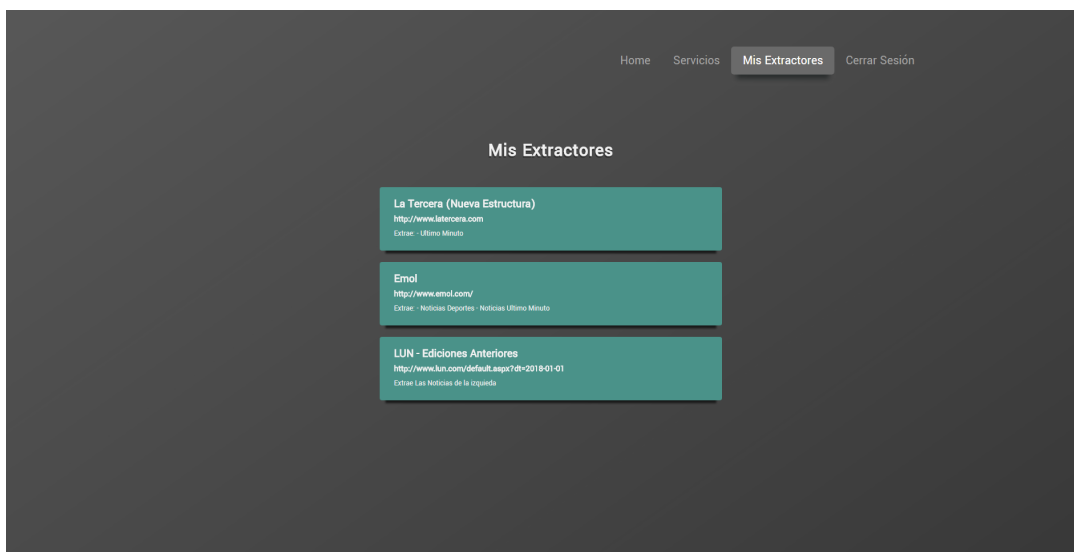


Figura 3.33: Página de listado de extractores del usuario.

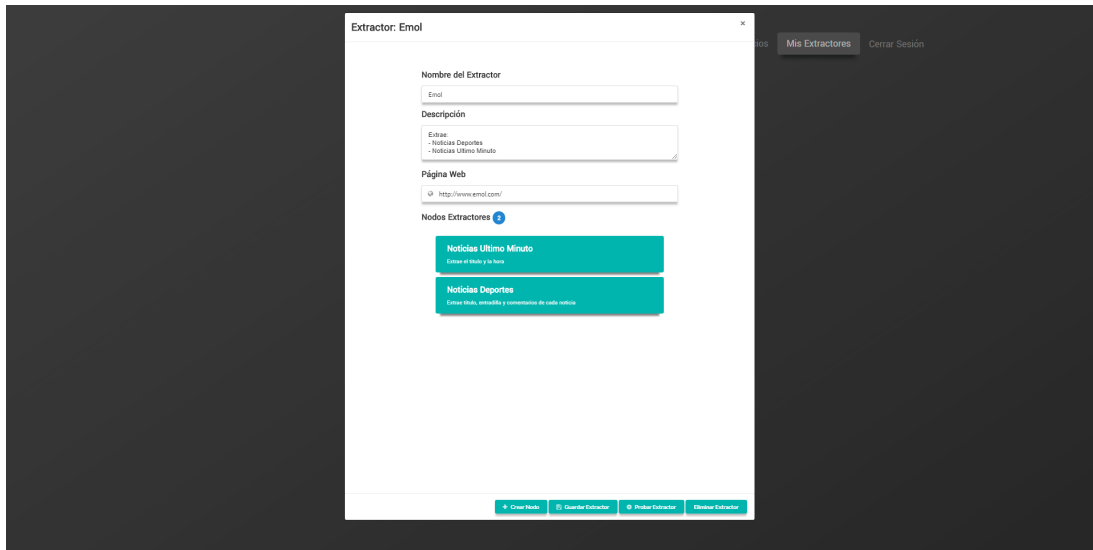


Figura 3.34: Página de visualización de las características de un extractor seleccionado.