UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN

DESARROLLO DE UNA HERRAMIENTA INTERACTIVA PARA LA CONSTRUCCIÓN DE UN “GROUND TRUTH” DE SEGMENTACIONES DE PÁGINAS WEB

Trabajo Especial de Grado presentado ante la ilustre
Universidad Central de Venezuela por

Br. García Lunardi, Jean Pearre

Tutor:
Dr. Sanoja, Andrés

Caracas - Venezuela

Mayo - 2018

ACTA de Veredicto

Quienes suscriben, miembros del jurado designado por el Consejo de la Escuela de Computación de la Facultad de Ciencias de la Universidad Central de Venezuela, para examinar el Trabajo Especial de Grado titulado "**Desarrollo de una Herramienta Interactiva para la Construcción de un Ground Truth de Segmentaciones de Páginas Web**" y presentado por el Bachiller Jean Pearre García Lunardi C.I: 23.868.370,a los fines de cumplir con el requisito legal para optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Leído el trabajo por cada uno de los Miembros del Jurado, se fijó el día 24 de Mayo de 2018, a las 11:00am, para que el autor lo defendiera en forma pública en el Aula PBIII de la Escuela de Computación, Facultad de Ciencias de la Universidad Central de Venezuela, lo cual realizó mediante una presentación oral de su contenido y luego respondió satisfactoriamente a las preguntas que le fueron formuladas por el Jurado, todo ello conforme a lo dispuesto en la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo con la nota de 20 puntos.

En fe de lo cual se levanta la presente Acta, en Caracas el día 24 de Mayo de 2018.

Prof. Andrés Sanoja - Tutor Firmante

Profa. Ana Leguizamo - Jurado Principal

Prof. Carlos Acosta - Jurado Principal

RESUMEN

El presente Trabajo Especial de Grado fue llevado a cabo con la finalidad de proveer un *ground truth* (base de la verdad) de segmentaciones manuales sobre una página Web para la posterior obtención de “la mejor segmentación”. Se establece entonces que el objetivo del TEG es el desarrollo de una herramienta Web que permita la construcción de una *ground truth* de segmentaciones de páginas Web.

La herramienta de segmentación manual se desarrolla en la forma de una extensión de Chrome, desarrollada con los lenguajes: Javascript, CSS3 y HTML5. La misma permite la generación de segmentaciones que son posteriormente guardadas en una base de datos Postgresql a través de una API RESTful (desarrollada en Python 3) conocida como MoB API, esto genera un *ground truth*, el cual hará posible la obtención de la “mejor segmentación” la cual podrá ser usada para la evaluación de los algoritmos de segmentación para que los mismos puedan ser adaptados a realizar la segmentación de una página Web desde el punto de vista de un usuario. .

Se hace uso del método Kanban como guía para la ejecución de actividades y se hace uso de la metodología de Desarrollo Adaptable de Software para la etapa de desarrollo de la herramienta. Durante la investigación previa se hizo evidente que los algoritmos de segmentación de página Web son de utilidad en la actualidad y poseen un amplio número de aplicaciones, además se evidencia que la mayor barrera para el desarrollo de estos algoritmos es la correcta evaluación de los mismos, por lo cual se demuestra la importancia del desarrollo de la herramienta planteada en este TEG.

Palabras claves: segmentación; página web; segmentación manual; ground truth; Javascript; Chrome; Python; Postgresql.

AGRADECIMIENTOS

A mi madre Jacqueline Lunardi y a mi padre Juan García por su incondicional apoyo y esfuerzo en mi formación como ser humano, por hacer hasta lo imposible para ofrecerme una excelente una excelente educación y su impartición de conocimientos y valores a lo largo de mi vida, y sobre todo por su abundante amor.

A mi hermana Estefanie García quien en innumerables ocasiones asumió el papel de mi segunda madre con sus cuidados y atenciones lo cual hizo posible el poder dedicar más de mi tiempo a los estudios, por soportarme en mis peores momentos y aun así seguir allí para mí, muchas gracias.

A mi tutor Andrés Sanoja, cuyo apoyo e instrucción fueron fundamentales para la realización de este trabajo, gracias por tenerme paciencia y explicarme el contenido de tantos años de trabajo con contagiante pasión, muchas gracias.

A la Universidad Central de Venezuela, mi casa de estudios, por permitir mi formación profesional en sus instalaciones.

A todos aquellos profesores y familiares que de alguna forma apoyaron mi crecimiento a lo largo de mis estudios, muchas gracias.

INTRODUCCIÓN

La WWW (World Wide Web) es un sistema de distribución de documentos de hipertexto o hipermedios interconectados y accesibles vía Internet.

La creación de la World Wide Web dió paso a la creación de sitios web, siendo este último un conjunto de páginas Web interconectadas entre sí por hipervínculos (*c.f* Capítulo 2.1.1).

La página Web es un documento digital de información accesible mediante un navegador de Internet, esta información se presenta generalmente en formato HTML, está compuesta por un conjunto de elementos ordenados en una estructura de árbol (el árbol DOM, *c.f* Capítulo 2.1.2), generado por el navegador a partir del código fuente HTML.

La segmentación es definida por la RAE (Real Academia Española) como el acto o consecuencia de segmentar (i.e dividir, formar segmentos o porciones). En el caso del presente trabajo investigativo, se habla de la segmentación de una página Web, la cual consiste en dividir dicha página Web en fragmentos llamados bloques, donde cada bloque representa distintos elementos de información en la página (*c.f* Capítulo 2.1.3).

La segmentación de una página Web es un proceso importante que puede ser aplicado en diferentes áreas (*c.f* las áreas de aplicación descritas al final de esta sección). Por la cantidad de páginas Web que deben ser segmentadas, la segmentación de la página debe ser realizada correctamente de forma automática, sin embargo, la correcta distribución de los segmentos dentro de una página es algo que puede resultar subjetivo dependiendo de la visualización que posea el usuario, por esto se debe evaluar el algoritmo de segmentación para comprobar su correcto funcionamiento. Esta evaluación no es posible realizarla en la actualidad de manera automatizada. El presente TEG facilita el llevar a cabo la evaluación de un algoritmo de segmentación al presentar una herramienta que permite a los usuarios realizar segmentaciones manuales de páginas Web, de forma cómoda, rápida e interactiva. Las segmentaciones generadas a partir de esta herramienta permiten la obtención de una *ground truth*, la cual representa una base de información obtenida mediante observación. Por medio de esta base se obtiene un elemento vital para la evaluación del algoritmo de segmentación, la “mejor segmentación”, la misma es utilizada como parámetro en la evaluación de un algoritmo de segmentación.

La segmentación de páginas Web tiene varias aplicaciones, las cuales exploramos para inspirarnos y conformar un dominio o contexto para nuestro trabajo. A continuación se describen algunas de ellas:

- **Procesos de SEO (Search Engine Optimization)**

  Es importante mantener la información accesible, de nada sirve tener un sitio web con información útil e importante si no se sabe cuál es su URL (Uniform Resource Locator) para poder acceder a él, es por esto que existen los Search Engine o “motores de búsqueda” (*e.g* Google) los cuales facilitan el proceso de búsqueda de los sitios web a lo largo de la red del internet. Sin embargo, realizar una búsqueda que arroje un resultado eficaz no es una tarea sencilla.

  Cada día se crean más sitios web, es por esto que los motores de búsqueda como Google aplican una serie de reglas para poder calificar las páginas Web y anexarlas a un *ranking* o clasificación que pueda ser usado al momento de arrojar los resultados de la búsqueda.

  Dicha evaluación debe ser hecha de forma automática, es decir, que debe ser posible analizar la página Web de forma automática para saber qué información se encuentra en ella y como está distribuida, es entonces cuando entra en acción la segmentación de la página Web.

  La segmentación de la página Web permite llevar a cabo un análisis del contenido de la página para que ésta pueda ser calificada y ubicada en un ranking.

- **Migración de Formatos**

  Con el constante desarrollo de las tecnologías, se puede obtener grandes mejoras que ofrecen diversas ventajas en el área donde se aplique la tecnología, sin embargo esto también trae consigo la obsolescencia de las versiones antiguas de esa tecnología que ha sido actualizada, los formatos de las páginas Web no son la excepción.

  Una página Web es un documento digital que se presenta generalmente en formato HTML. El formato HTML ha ido evolucionando a lo largo de los años, actualmente la última versión es HTML5, sin embargo aun existen páginas web con formato HTML4. Como es de esperarse, el formato HTML5 posee ventajas

con respecto al HTML4, por esto mismo se crea la necesidad de querer llevar las páginas que se encuentran en formato HTML4 al HTML5, es aquí donde entra en acción la segmentación de la página Web.

La segmentación de la página Web en formato HTML4 permite la creación de una nueva página Web usando el formato HTML5 al poder identificar los segmentos de la página anterior, esto es evidenciable en el trabajo de Andrés Sanoja y Stéphane Gançarski (*c.f* Capítulo 2.3).

- **Archivamiento de la Web (Web Archiving)**

Es el proceso mediante el cual se coleccionan páginas Web para asegurar la preservación de las mismas para futuras búsquedas, historiadores y el público. Generalmente se emplean los *Web Crawlers* o arañas para la captación automática de la información dado el gran tamaño de la Web.

Dichas arañas hacen capturas de la información existente en una página Web y lo almacenan, esto deben hacerlo cada cierto tiempo para asegurar que la información que poseen esté actualizada, aquí entra en juego la gran cantidad de espacio de almacenamiento requerido, debido a la gran cantidad de páginas Web, es por esto que las arañas deben priorizar la descarga y almacenamiento de aquellas páginas web que realmente hayan sufrido algún cambio, es entonces cuando entra en acción la segmentación de la página Web.

La segmentación de la página Web permite comparar dos versiones de la misma página Web (la versión que actualmente se tiene almacenada y la versión que se planea almacenar) y encontrar las diferencias entre ellas, esto permite detectar si vale la pena descargar y almacenar esa versión.

- **Bloqueo de Contenido (Content Blocking)**

El bloqueo de contenido, como su nombre lo indica, consiste en bloquear ciertos contenidos no deseados que se encuentren en la página Web para impedir su visualización.

Es mayormente usado por herramientas, complementos o extensiones de navegadores para poder bloquear las publicidades de una página Web.

La segmentación de la página Web permite poder identificar estos segmentos que poseen el contenido no deseado para que pueda ser bloqueado.

Esta es una práctica que no es bien vista por los dueños de los sitios web, dado que la publicidad en muchos casos es el principal ingreso del sitio, sin embargo, es innegable admitir que en la actualidad es preciso e imprescindible poseer alguna de estas herramientas de bloqueo (*e.g* Adblock Plus) para evitar la sobresaturación de publicidad existente, la cual no solo resulta molesta sino también potencialmente peligrosa para nuestras computadoras.

Durante el proceso de desarrollo del sistema planteado, se fueron modificando ciertos aspectos que habían sido considerados en un principio durante la etapa de investigación. Como es el caso de cambiar el objetivo “Proponer una técnica de validación social la cual permita la evaluación de los resultados obtenidos por un grupo de usuarios que segmentan manualmente una misma página Web“ por “Procesar los datos resultantes de un grupo de usuarios que segmentan manualmente una misma página Web para la conformación de la mejor segmentación“ dado que se consideró el hecho de que la validación social requeriría de la intervención de los usuarios, cosa que se quería evitar pues ya de por si la actividad de segmentar manualmente una página Web puede consumir una porción considerable de tiempo, por lo que se deseaba eliminar al usuario el mayor número de tareas posibles. En todos estos casos se conserva la propuesta inicial y la propuesta final junto con la justificación de la misma.

TABLA DE CONTENIDO

RESUMEN 3

AGRADECIMIENTOS 4

INTRODUCCIÓN 5

TABLA DE CONTENIDO 9

ÍNDICE DE FIGURAS 11

ÍNDICE DE TABLAS 13

CAPÍTULO I: EL PROBLEMA 14

1. Planteamiento y Delimitación del Problema 14
2. Justificación e Importancia 15
3. Descripción de la Solución 15
4. Objetivos de la Investigación 19
   - 4.1. Objetivo General 19
   - 4.2. Objetivos Específicos 19
5. Justificación e Importancia 20
6. Alcance 20

CAPÍTULO II: MARCO TEÓRICO 21

1. Definición de Términos 21
   - 1.1. Página Web 21
   - 1.2. DOM (Document Object Model) 22
   - 1.3. Segmentación de Páginas Web 23
   - 1.4. SPA (Single-Page Application) 24
   - 1.5. Usabilidad 25
   - 1.6. API RESTful 25
   - 1.7. La Mejor Segmentación Basada en Popularidad 26
   - 1.8. Herramientas Tecnológicas 26
2. Antecedentes 29
3. Trabajos de Referencia 34
   - 3.1. BoM 35
   - 3.2. MoB 37
   - 3.3. Proyecto SCAPE 40
4. Trabajos Relacionados 40

CAPÍTULO III: MARCO METODOLÓGICO 43

1. Metodología de Trabajo Kanban 43
2. Desarrollo Adaptable de Software 44

CAPÍTULO IV: DESARROLLO DE LA SOLUCIÓN **47**
1. Descripción general de la solución 47
2. Arquitectura de la solución 48
3. Ciclos de Desarrollo 49
   3.1. 1er Ciclo: Extensión MoB 50
   3.2. 2do Ciclo: MoB API 71
   3.3. 3er Ciclo: MoB Repository (Repositorio MoB) 76
   3.4. 4to Ciclo: La Mejor Segmentación 97
   3.5. 5to Ciclo: Mejoras 101
4. Pruebas de aceptación 103
   4.1. Pruebas funcionales 104
   4.2. Pruebas no funcionales 106

CAPÍTULO V: CONCLUSIONES **117**
1. Contribución 118
2. Recomendaciones 119
3. Trabajos Futuros 119

ANEXOS **120**
1. Instalación del Ambiente 120
   1.1. Configuración del Servidor 120
   1.2. Para el Sistema Manejador de Base de Datos 121
   1.3. Últimos ajustes 122
2. Documentación de MoB API 123

REFERENCIAS BIBLIOGRÁFICAS **133**

ÍNDICE DE FIGURAS

Figura 1: Ejemplo de la herramienta Pagelyzer……..…32
Figura 2: Proceso de segmentación realizada por BoM……..…….36
Figura 3: Segmentación automática realizada por BoM……..…….37
Figura 4: Segmentación previa realizada por BoM…….……38
Figura 5: Cuadro de diálogo…….……..39
Figura 6: Bloque segmentado…….……..……..…....39
Figura 7: Ejemplo de la herramienta FitLayout…….……..…….41
Figura 8: Ejemplo de la herramienta Chrome DevTools…….……...42
Figura 9: Ciclo de la metodología de desarrollo ASD…….……44
Figura 10: Modelo de la propuesta de TEG…….……..……...48
Figura 11: Modelo de la Arquitectura MVC del Sistema…….……...48
Figura 12: Diagrama de Casos de Uso de la Extensión MoB, Nivel 1……..51
Figura 13: Diagrama de Casos de Uso 6 de la Extensión MoB, Nivel 2……...52
Figura 14: Mock-up Extensión de MoB…….……..…59
Figura 15: Herramienta y Paleta de Colores…….…….....59
Figura 16: Acción de Crear Bloque…….……...……...64
Figura 17: Acción Eliminar Bloque…….……...……...64
Figura 18: Acción Unir Bloques…….……...……...65
Figura 19: Acción Cortar Bloques…….……..66
Figura 20: Acción Etiquetar Bloque…..……..……..66
Figura 21: Acción Seleccionar Bloque…….……..…….67
Figura 22: Acción Panel de Información……..…….67
Figura 23: Comparación de herramientas MoB…….....70
Figura 24: Modelo Entidad-Relación de las Tablas en el Sistema……..….72
Figura 25: Diagrama de Casos de Uso para el componente Mob Repository, Nivel 1…….77
Figura 26: Diagrama de Casos de Uso 3 para el componente Mob Repository, Nivel 2…….78
Figura 27: Diagrama de Casos de Uso 7 para el componente Mob Repository, Nivel 2…….79
Figura 28: Diagrama de Casos de Uso 8 para el componente Mob Repository, Nivel 2…….79
Figura 29: Diferencia de Alertas..……...102
Figura 30: Interfaz Extensión MoB...……102
Figura 31: Interfaz Repositorio de MoB..……..103

Figura 32: Pregunta 1 del Cuestionario.……..……..107
Figura 33: Pregunta 2 del Cuestionario.……107
Figura 34: Pregunta 3 del Cuestionario..……108
Figura 35: Pregunta 4 del Cuestionario..……...108
Figura 36: Pregunta 5 del Cuestionario..…….....109
Figura 37: Pregunta 6 del Cuestionario..……...……110
Figura 38: Pregunta 7 del Cuestionario..……..…110
Figura 39: Pregunta 8 del Cuestionario..…….……111
Figura 40: Pregunta 9 del Cuestionario..……...……112
Figura 41: Pregunta 10 del Cuestionario..……...112
Figura 42: Pregunta 11 del Cuestionario...……..……113
Figura 43: Pregunta 12 del Cuestionario..……..……114
Figura 44: Pregunta 13 del Cuestionario..……..……114
Figura 45: Pregunta 14 del Cuestionario.……...115

ÍNDICE DE TABLAS

Tabla 1: Extensión MoB UC.1……..52
Tabla 2: Extensión MoB UC.2……..53
Tabla 3: Extensión MoB UC.3…….....54
Tabla 4: Extensión MoB UC.4……...……...55
Tabla 5: Extensión MoB UC.5……...……...56
Tabla 6: Extensión MoB UC.6……..57
Tabla 7: Repositorio MoB UC.1……...80
Tabla 8: Repositorio MoB UC.2……...80
Tabla 9: Repositorio MoB UC.3……...81
Tabla 10: Repositorio MoB UC.3.1……..82
Tabla 11: Repositorio MoB UC.3.2……..83
Tabla 12: Repositorio MoB UC.3.2.1……..84
Tabla 13: Repositorio MoB UC.3.2.1.1……...85
Tabla 14: Repositorio MoB UC.3.2.1.2……...86
Tabla 15: Repositorio MoB UC.3.2.1.3……..87
Tabla 16: Repositorio MoB UC.3.2.1.3.1……...87
Tabla 17: Repositorio MoB UC.4…….....88
Tabla 18: Repositorio MoB UC.5……89
Tabla 19: Repositorio MoB UC.6…….....90
Tabla 20: Repositorio MoB UC.7…….....91
Tabla 21: Repositorio MoB UC.7.1……..91
Tabla 22: Repositorio MoB UC.7.1.1……..92
Tabla 23: Repositorio MoB UC.8……93
Tabla 24: Repositorio MoB UC.8.1……..94
Tabla 25: Repositorio MoB UC.9……95
Tabla 26: Prueba Caja Negra de Extensión MoB……...104
Tabla 27: Prueba Caja Negra de Repositorio MoB……105

CAPÍTULO I: EL PROBLEMA

1.1. Planteamiento y Delimitación del Problema

Los sitios web comenzaron siendo simples colecciones de archivos (páginas Web) interconectados mediante vínculos Web. En la actualidad las páginas Web se han desarrollado al punto donde no sólo están compuestas por texto e hipervínculos sino por otros tipos de datos más complejos, sin embargo, la estructura de la página sigue siendo virtualmente la misma: un conjunto de elementos relacionados entre sí mediante una estructura de árbol (*i.e*. el árbol DOM).

El crecimiento exponencial de las páginas Web en la actualidad dificulta la tarea de los buscadores como Google en poder indexar las nuevas páginas a sus resultados, se necesita poder hacer un análisis mucho más rápido de los contenidos de cada página, se necesita poder separar la información de utilidad de la que no lo es (*c.f.* Capítulo 2.1.3).

Tener la capacidad de conocer los segmentos o bloques de contenido de una página Web es una necesidad que está cada vez más presente en la actualidad, ya que tal control sobre la página puede permitir procesos de extracción de información, migración de formatos, control de versiones, entre otros.

En la actualidad, hasta donde conocemos, no se cuenta con una base para realizar análisis comparativo de algoritmos de segmentación de páginas Web. Cada desarrollador utiliza un modelo de evaluación *ad hoc* (a la medida) y ajustado a sus necesidades. Esto hace que los resultados de las evaluaciones sean subjetivas. Es necesario una manera de realizar estas comparaciones de la manera más objetiva posible.

La comparación puede realizarse mediante la comparación de los árboles de segmentación y los rectángulos obtenidos, independientemente del algoritmo.

1.2. Justificación e Importancia

Para poder medir la calidad de una segmentación debemos tener algo contra qué comparar. En el campo de estudio, se han hecho avances, pero no se cuenta con una base de información (*ground truth*) confiable y completa contra la cual comparar.

En otras palabras, es importante poder obtener la salida de la segmentación (*c.f.* Capítulo 2.1.3) para que pueda ser comparado con el resultado arrojado por un algoritmo de segmentación automático. Esto con la finalidad de poder verificar si el algoritmo de segmentación funciona de forma correcta o no. A su vez es de vital importancia para el desarrollo de algoritmos de segmentación de páginas Web, ya que permitirá a los programadores realizar sus pruebas.

La segmentación manual de una página Web puede resultar ser un proceso complejo, debido a que se requieren conocimientos de HTML y la estructura de los del árbol DOM, es por esto que esta investigación no sólo busca desarrollar la herramienta que permita llevar a cabo la segmentación manual junto con todo lo anteriormente descrito, sino que además busca reducir la complejidad que se le puede presentar al usuario a la hora de segmentar la página Web, ofreciendo una interfaz usable que cumpla con los parámetros de IHC necesarios para que cualquier usuario (teniendo o no conocimientos de HTML) pueda realizar la segmentación de forma rápida y cómoda.

1.3. Descripción de la Solución

Se realizará el desarrollo de una herramienta Web, específicamente una extensión de un explorador Web. En particular para el explorador Chrome. Esta extensión debe permitir a los usuarios realizar una segmentación manual de una

página Web. Como resultado se obtiene la representación de dicha segmentación en un árbol de segmentación y un conjunto de rectángulos (*c.f.* *Capítulo 2.1.3*, *Capítulo 2.3.1*). Esta es la información básica necesaria para construir un *ground truth.* Se considera explorar otras formas de bloques aparte de las rectangulares.

La construcción del *ground truth* pasa por dos etapas: segmentación manual en el cliente y selección de la "mejor" segmentación en el servidor. La mejor segmentación es el "acuerdo" al que se llega cuando se crea una segmentación con todas las características compartidas entre todas las segmentaciones que conforman el *ground truth*.

En el cliente:

La extensión utiliza el algoritmo de Block-o-Matic (BoM) para realizar una segmentación previa antes de darle el control al usuario. La interfaz ofrecerá la funcionalidad de poder agregar, editar y eliminar un bloque, a la vez que puede asignarle una etiqueta para identificarlo, la herramienta le indicará al usuario en todo momento cuál es el bloque en el que ejerce la acción. Este procedimiento puede ser realizado por varios usuarios sobre una misma página.

Una vez el usuario considere que la segmentación se ha completado (y la misma cumpla con los requisitos de granularidad establecidos), enviará el resultado al servidor, el servidor tomará dichos datos y los almacenará en una base de datos para su posterior recuperación y análisis.

Los procesos mencionados anteriormente podrían verse como la etapa de "captación de datos", donde se producen los datos necesarios para un futuro análisis.

En el servidor:

La etapa de análisis es igual de importante, ya que es la que permitirá la evaluación del correcto funcionamiento de un algoritmo de segmentación. Para la etapa de análisis se siguen los siguientes pasos:

1. Se toman todos los resultados de segmentación de una página dada y se saca un promedio de ellos, esto es para obtener la segmentación ideal. Para ello se utilizan técnicas de social validation (*c.f.* Sección III-A-7).
2. Se toma el resultado del árbol de segmentación arrojado por el algoritmo de segmentación y se compara con el promediado de los resultados obtenidos por los usuarios.
3. Se toma el árbol estructural de bloques y las etiquetas creadas por el algoritmo de segmentación y se compara con el promediado de los resultados obtenidos por los usuarios.
4. En los puntos 2 y 3 se obtiene un puntaje, que indicará que tan cercano estuvo el algoritmo de segmentación en la segmentación de la página según la segmentación visualizada por un usuario promedio.

Debido a que la actividad de segmentación es tediosa se incorpora un sistema de puntajes, para añadir competencia y motivar a los usuarios a segmentar las páginas Web.

Los puntajes obtenidos por el usuario dada la segmentación de una página Web cualquiera están vinculados con el tiempo promedio en que se tardó en realizar la segmentación en dicha página, que tan cerca del resultado promedio está la segmentación realizada y cuán completa es. Esta "cercanía" se estima mediante el cálculo de la frecuencia de ocurrencia de cada bloque, en el conjunto de segmentaciones. Por ejemplo: si una cantidad k de usuarios marcaron el bloque i de manera similar existe la certeza $C(i)=k/n$ (n es la cantidad total de usuarios) que el bloque sea uno correcto (por consenso). Adicionalmente si un usuario consigue realizar la segmentación en menos tiempo que el promedio de los otros usuarios, y cada $C(i)$ es mejor que un promedio, se le asignan más puntos a su segmentación. Se debe considerar también cuán completa es la segmentación. Para ello se suman las áreas de los rectángulos y se comparan con el área de la página. Si es significativamente menor pierde puntos, en caso contrario los gana. Esta restricción busca determinar si la segmentación realizada presenta una partición de la página, o por el contrario determinar si hay regiones de la página sin segmentar. Finalmente se actualiza una

tabla con los puntajes. Es importante mencionar que la cantidad de usuarios mínimos para que un resultado sea significativo, debe ser estimado de manera empírica y forma parte de las actividades del TEG.

Ejemplo: un usuario segmenta una página que posee 7 bloques con un tiempo promedio de segmentación de 2 minutos y cubre toda la página. En este momento es el usuario con puntaje máximo y mejor valores de C(i). Otro usuario indica 3 bloques realiza la segmentación en 30 segundos pero no marca el *footer* de la página. Dos de los bloques tienen *C(i)* similar. Este último, en teoría, obtiene puntos ya que tuvo mayor precisión en menor tiempo. Sin embargo su segmentación no es una partición de la página, por lo que pierde puntos. La meta es fomentar que el usuario realice una segmentación rápida y eficaz, no bastará con solo segmentar al azar, sino que deberá segmentar de forma correcta (o al menos lo que el promedio indique como correcto).

Se llevará un sistema de puntuación, también se llevará un control de los usuarios junto con sus respectivas segmentaciones y puntuaciones asociadas.

Antes de realizar la segmentación, el usuario debe registrarse e identificarse en el sistema, una vez realizada la segmentación y enviada al servidor, se le otorga una puntuación y tanto la segmentación realizada como el puntaje obtenido son almacenados en la base de datos (ambas asociadas al usuario). Si el usuario desea, en un futuro, volver a segmentar la misma página, los datos de segmentación en la base de datos son actualizados, así como su puntaje.

La interfaz permitirá mostrar una sección con la lista de las 10 mejores puntuaciones obtenidas por la segmentación de la página, donde aparecerá el puntaje junto con el nombre de usuario del que realizó la segmentación.

La mejor segmentación en la *ground truth* será utilizada por una herramienta de evaluación, la cual no forma parte del alcance de este trabajo.

1.4. Objetivos de la Investigación

1.4.1. Objetivo General

Desarrollar una herramienta Web que permita la segmentación manual de una página Web, su almacenamiento y análisis, para conformar una *ground truth*.

1.4.2. Objetivos Específicos

- Desarrollar funcionalidades de agregar, editar, eliminar y etiquetar bloques de segmentos para permitir al usuario segmentar visualmente una página web.
- Desarrollar una funcionalidad que permita tomar la segmentación visual realizada por el usuario y transformarla en un árbol de segmentación basado en el árbol DOM de la página web segmentada.
- Desarrollar una funcionalidad que permita tomar la segmentación visual realizada por el usuario y obtener el árbol estructural de bloques (rectangulares) asociados al diseño de la página.
- Implementar funcionalidades de registro e identificación de usuarios, las cuales permitan el control y manejo de sesión de usuarios.
- Implementar una Base de Datos que almacene todos los resultados obtenidos por la herramienta y los datos de los usuarios que hacen uso de ella.
- Implementar funcionalidades en el lado del servidor tal que permitan el análisis y la visualización de la data almacenada.
- Mejorar las vistas de la herramienta para que esta herramienta cumpla con los requerimientos de IHC y sea una herramienta usable.
- Desarrollar un sistema de recompensas y puntaje que vuelva interactivo y entretenido el proceso de segmentación.
- Realizar pruebas dinámicas de aceptación de caja negra para evaluar el correcto funcionamiento de la herramienta.
- Conformar un *ground truth* a partir de los datos obtenidos por la herramienta de segmentación manual.
- Proponer la mejor segmentación dentro de un grupo de usuarios que segmentan manualmente una misma página Web.

1.5. Justificación e Importancia

Para poder medir la calidad de una segmentación debemos tener algo contra qué comparar. En el campo de estudio, se han hecho avances, pero no se cuenta con una base de información (*ground truth*) confiable y completa contra la cual comparar.

En otras palabras, es importante poder obtener la salida de la segmentación (*c.f.* Capítulo 2.1.3) para que pueda ser comparado con el resultado arrojado por un algoritmo de segmentación automático. Esto con la finalidad de poder verificar si el algoritmo de segmentación funciona de forma correcta o no. A su vez es de vital importancia para el desarrollo de algoritmos de segmentación de páginas Web, ya que permitirá a los programadores realizar sus pruebas.

La segmentación manual de una página Web puede resultar ser un proceso complejo, debido a que se requieren conocimientos de HTML y la estructura del árbol DOM. Esta investigación no sólo busca desarrollar la herramienta que permita llevar a cabo la segmentación manual, sino que además busca reducir la complejidad que se le puede presentar al usuario a la hora de segmentar. Se ofrece una interfaz usable que cumpla con los parámetros de IHC necesarios para que cualquier usuario (teniendo o no conocimientos de HTML) pueda realizar la segmentación de forma rápida y precisa.

1.6. Alcance

Para el desarrollo del TEG se tienen los siguientes límites:

- Solo se considerara la versión Desktop de Google Chrome y Chromium browser.
- Se considerarán páginas Web que hayan sido desarrollada con el lenguaje HTML en cualquiera de sus versiones (desde HTML1 hasta HTML5).
- El desarrollo cubre la conformación de la *ground truth*. Se indica la mejor segmentación pero no se incluye la evaluación.
- Las estimaciones de parámetros se indicarán de manera empírica.

CAPÍTULO II: MARCO TEÓRICO

En este capítulo se presentan los conceptos utilizados en la presente TEG. En la sección 2.1 se describirán conceptos relacionados con las páginas Web, la segmentación, la usabilidad y las tecnologías de interés a ser utilizadas. En la sección 2.2 se presentan los antecedentes. En la sección 2.3 se presentan los trabajos de referencia. En la sección 2.4 se presentan trabajos relacionados.

2.1. Definición de Términos

En esta sección se describen los conceptos fundamentales para la realización del TEG. En la sección 2.1.1 se describen las páginas Web. En la sección 2.1.2 se describe el DOM. En la sección 2.1.3 se describe la segmentación de páginas Web. En la sección 2.1.4 se presenta la técnica de SPA (Single-Page Application) la cual nos guiará en la organización de los elementos de la aplicación. En la sección 2.1.5 se describe la usabilidad. En la sección 2.1.6 se presenta las API-RESTful, pensadas como mecanismo de comunicación entre los componentes de la aplicación. En la sección 2.1.7 se describe la Social Validation (Validación Social) importante para la selección de una mejor segmentación realizada por un grupo de usuarios. Finalmente en la sección 2.1.8 se describen las herramientas tecnológicas que se plantean ser usadas en el desarrollo de la herramienta.

2.1.1. Página Web

Según González y Cordero (2001, p.20), una página Web es una fuente de de información adaptada para la World Wide Web (WWW) y accesible mediante un navegador de Internet. Esta información se presenta generalmente en formato HTML y puede contener hiperenlaces a otras páginas Web, constituyendo la red enlazada de la World Wide Web.

Un sitio web es un conjunto de páginas Web interconectadas por hipervínculos. Cada página Web posee una única URL (Uniform Resource Locator).

Es de interés destacar que la herramienta a desarrollar trabajará sobre una página Web específica y no en un sitio web.

Las páginas Web pueden estar conformadas por reglas de CSS. Estas reglas aportan características estéticas. Pueden utilizarse *scripts* (*e.g.* Javascript) los cuales le aportan funcionalidades y eventos.

Una página puede componerse utilizando diferentes tipos de componentes. Ejemplos de estos componentes son: imágenes, sonido, incluso aplicaciones embebidas (*e.g.* una aplicación que proporcione una sala de chat para tu página web). En el contexto de páginas Web, todo es representado como un Elemento. Estos elementos forman parte de una estructura de un árbol (el árbol DOM), generado por el navegador a partir del código fuente HTML.

Estos elementos se estructuran en base a las normas creadas por organismos como el World Wide Web Consortium (W3C). Establecen directivas con la intención de normalizar el diseño, y así servir de guía a los desarrolladores de cómo debe procesarse (en inglés *rendering*) un documento HTML en un navegador.

Son de uso común varios términos para referirse a las páginas Web: página, documento, documento Web, entre otros.

2.1.2. DOM (Document Object Model)

Según la W3C (2005), el DOM representa una interfaz de plataforma y lenguaje neutral que le permite a programas y *scripts* acceder dinámicamente a los documentos y actualizar su contenido, estructura y estilo. El documento puede estar escrito en HTML, XHTML o XML.

Sin importar el lenguaje de marcado, la forma de acceder a los elementos no cambia pues el DOM aporta un modelo estándar para poder acceder a cada uno de los elementos dentro del documento y manipularlos.

El DOM permite la posibilidad de que acceder y modificar la estructura y el diseño de los documentos de forma dinámica a través de lenguajes como ECMAScript (JavaScript) proporcionando funcionalidad y dinamismo a la página Web.

Para el presente TEG el manejo del DOM es de vital importancia. La herramienta interactúa directamente con la versión procesada (renderizada) en un navegador de una página Web (*i.e.* el DOM). El DOM es uno de los insumos principales para realizar una segmentación (*c.f.* Sección 2.1.3) y su manipulación.

2.1.3. Segmentación de Páginas Web

La segmentación de una página Web consiste en dividir dicha página Web en fragmentos coherentes[1] llamados bloques. Cada bloque representa distintos elementos de información en la página.

La segmentación de páginas Web es de vital importancia para los motores de búsqueda como Google. Es utilizada para calcular la importancia y relevancia del contenido de una página para la Web. Permite reducir el "ruido" o información sin importancia dentro de la página a analizar.

El ruido de la página Web son todos aquellos bloques de contenido que no se relacionan con el tema del contenido principal que se está analizando (*e.g.* publicidad, redes sociales, búsqueda, elementos de navegación para la generación de tráfico). Este tipo de información no aporta a la búsqueda, por lo que se debe ignorar. El ruido más común proviene de las publicidades, ya que pueden provocar resultados erróneos al entremezclarse con el contenido de la página, y por esto es importante ubicarlas y separarlas.

La técnica de segmentación de una página Web (junto con otras medidas y técnicas[2]) permite que la página adquiera un mejor posicionamiento en los motores de búsqueda. Esta técnica se conoce como SEO (Search Engine Optimization) u Optimización de Motores de Búsqueda (Cai, 2004). En la actualidad el SEO es de vital importancia para una empresa que desea darse a conocer, es lo que hace la diferencia de que pueda ser encontrada fácilmente en la Web (y en consecuencia tener un mayor flujo de usuarios y visitantes) o que quede opacada por los otros cientos de resultados.

La segmentación de la página Web no solo es usada por los motores de búsqueda, existen otras aplicaciones donde es importante conocer los contenidos de la

[1] Por coherente se refiere a que cada bloque debe tener un sentido para un usuario, *e.g.*: el menú de navegación de una página Web debe ser considerado como un único bloque.
[2] Como por ejemplo usar palabras claves que sean relevantes del tema dentro de los contenidos, y usar links que redirigen a otras páginas que poseen un tema similar.

página y la distribución de los mismos, como por ejemplo, la migración de formatos, el archivamiento de la Web y el bloqueo de contenido (*c.f.* Introducción) .

Este TEG busca proporcionar una herramienta de segmentación manual que pueda ser usada por un usuario cualquiera para segmentar una página Web, todo esto con la finalidad de obtener una salida que representará una segmentación manual de la página Web, esta salida será usada en un análisis en conjunto con otras versiones de la salida (la misma página segmentada por diferentes usuarios) lo cual generará una salida promedio, es decir, una versión óptima de cómo debería ser el árbol DOM de la página segmentada en orden de que esté correctamente segmentada, es necesario aplicar estos análisis debido a que la “correcta” o “incorrecta” segmentación de una página web puede ser subjetiva a la vista del usuario.

2.1.4. SPA (Single-Page Application)

Un *single-page application*, o aplicación de página única es una aplicación Web la cual está conformada por una sola página Web, con el propósito de permitir a los usuarios interactuar con una aplicación Web como si fuese una aplicación de escritorio. En un SPA todos los recursos (*e.g.* HTML, JavaScript, CSS) pueden cargarse de dos maneras: junto con la página o de manera asíncrona con el uso de AJAX (Asynchronous JavaScript And XML). La carga asíncrona es útil cuando se necesita cargar contenido como respuesta de las acciones del usuario.

Para el desarrollo de la herramienta se tiene la intención de organizar los componentes como un SPA. Así poder usar la ventaja que ofrece una SPA: mayor rapidez en el cliente al poder controlar las vistas desde el mismo cliente cargando dinámicamente cualquier recurso o vista que se necesite con AJAX. Como resultado, el usuario obtiene la sensación de que interactúa con una aplicación de escritorio cuando hace uso de la herramienta.

2.1.5. Usabilidad

La **IHC** (Interacción Humano-Computador) es un área de la informática que se encarga de estudiar la forma en que los seres humanos interactúan con artefactos, sistemas o infraestructuras computacionales.

La **usabilidad** es definida por la ISO (International Organization for Standardization) en su estándar ISO 9126 (1991) como:

"*La usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso.*"

La usabilidad viene representando una característica que posee una herramienta de software, la cual mide la facilidad que tiene el usuario en interactuar con la misma, se dice que un software es usable cuando el mismo presenta claridad y elegancia, permitiendo al usuario conocer al instante lo que debe hacer para lograr su objetivo.

Para el desarrollo del presente TEG, se busca presentar una interfaz usable mediante la cual el usuario pueda segmentar la página Web de forma rápida y precisa. También se desea aplicar soporte de idiomas al permitir el cambio de idioma dentro de la herramienta, permitiendo a usuarios de diferentes culturas hacer uso del software.

2.1.6. API RESTful

La interfaz de programación de aplicaciones, abreviada como API del inglés: Application Programming Interface, es un conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Un RESTful API, como se describió anteriormente, es un servicio que funciona como un estándar para compartir información, en un sistema de Consulta y Respuesta (Request -> Response).

La arquitectura REST (del inglés: Representational State Transfer) trabaja sobre el protocolo HTTP. Por consiguiente, los procedimientos o métodos de comunicación son los mismos que HTTP, siendo los principales: GET, POST, PUT, DELETE. Otros

métodos que se utilizan en RESTful API son OPTIONS y HEAD. Este último se emplea para pasar parámetros de validación, autorización y tipo de procesamiento, entre otras funciones.

En la presente investigación se usará una API RESTful para ser de intermediaria entre el cliente y la base de datos.

2.1.7. La Mejor Segmentación Basada en Popularidad

En la etapa de investigación se consideró obtener la mejor segmentación a través del enfoque de la “validación social”, sin embargo, dicho enfoque requiere del tiempo y atención de los usuarios, cuando en realidad se desea reducir en lo posible las tareas del usuario. Seguir por el enfoque de la validación social hubiese ocasionado justamente lo contrario, a medida que el *ground truth* crezca (lo cual es esencial para una “mejor segmentación” óptima). También incrementa el tiempo que el usuario debe invertir en dar su opinión sobre cada segmentación. Es por esto que se decide realizar un enfoque más automatizado. Bajo la misma idea de que se debe llegar a un “acuerdo” entre todos los usuarios, es decir, una especie de promedio entre todas las segmentaciones de los usuario, se toma el *ground truth* de una página Web específica y se construye desde cero una segmentación basada en todas las características compartidas por más de la mitad de las segmentaciones (es decir, la mayoría), a esta segmentación se la conoce como “la mejor segmentación”.

2.1.8. Herramientas Tecnológicas

Se presentan ahora un conjunto de tecnologías mediante las cuales se desarrolla la herramienta que da solución a la problemática.

La solución está basada en un enfoque Web, como una extensión para el navegador Chrome/Chromium, además cuenta con una interfaz para la visualización de las segmentaciones almacenadas. A continuación se listan las herramientas utilizadas clasificadas en: diseño del cliente, funcionalidad del cliente, funcionalidad del servidor y almacenamiento.

Diseño del Cliente

Estas herramientas controlan la estructura y estilos estéticos de las vistas de la interfaz de usuario en la aplicación, esto representa una parte importante en cuanto a los aspectos de Interacción Humano-Computador necesarios para que la aplicación sea usable.

- **HTML5:** HyperText Markup Language, es un lenguaje de marcado para la elaboración de documentos HTML (páginas Web). Describe la estructura y el contenido semántico de un documento Web. Permite estructurar el contenido de las vistas de la extensión y el sitio Web donde se podrán visualizar las segmentaciones (el Repositorio MoB).

- **CSS3:** Cascading Style Sheets, o hoja de estilo en cascada, es un lenguaje de hoja de estilos usada para describir la semántica de presentación (la vista y formato) de un documento escrito en un lenguaje de marcado. Su aplicación más común es darle estilos a páginas Web escritas en HTML y XHTML. Permite darle estilos estéticos a los contenidos estructurados en las diferentes vistas desarrolladas.

Funcionalidad del Cliente

Estas herramientas ayudan a controlar el comportamiento de la aplicación en el lado del cliente, permitiéndole al usuario realizar acciones como: agregar un bloque, identificar un bloque o borrar un bloque. Toda esta información se mantiene y se controla en el lado del cliente para que una vez el usuario finalice la segmentación puedan ser enviados al servidor.

- **Javascript:** Es un lenguaje de programación interpretado, es orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza ampliamente del lado del cliente para mejorar la experiencia del usuario con las páginas Web, haciéndolas más dinámicas e interactivas. En el caso del presente trabajo se utiliza principalmente para otorgar funcionalidades a la extensión con la que interactúa el usuario para realizar la segmentación de la página Web.

- **Jquery:** Es una librería de Javascript que ayuda a reducir las lineas de codigos de Javascript, permitiendo el correcto funcionamiento del código a través de los diferentes navegadores.

Existen diferentes frameworks basados en Javascript los cuales facilitan la implementación de funcionalidades en el lado del cliente, como por ejemplo: Angularjs2 y Emberjs; ambos estudiados durante el seminario como alternativas para programar la extensión. Sin embargo, durante la etapa de desarrollo del presente trabajo se evidenció que era suficiente utilizar Javascript puro (*aka* Vanilla Javascript) sin perder capacidad funcional. Como ventaja se evita *overhead* o complejidad adicional innecesaria por utilizar una librería externa.

Funcionalidad del Servidor

Esta herramienta permite controlar los datos obtenidos de la segmentación, procesarlos y enviarlos al almacén de datos.

Durante la etapa de investigación del presente TEG, se decidió usar el framework Django mayormente por su lenguaje de programación (Python), sin embargo, más tarde se concluyó que para realizar un API RESTful en Python no era necesario incluir un framework tan grande como Django, por este motivo se decide usar Flask, el cual es un micro framework de Python. El término “micro” no quiere decir que carezca en funcionalidades, solo expresa que Flask mantiene un núcleo fuerte pero extensible, es decir, que cualquier funcionalidad extra que se desee puede ser añadida más adelante.

Python es un lenguaje de programación interpretado, su filosofía está basada en hacer desarrollos de software con un código visible, por esto mismo obliga al usuario a indentar correctamente el código y omite las señales de fin de línea como el punto y coma en Javascript. Soporta programación orientada a objetos, imperativa y funcional.

Esto permite poder desarrollar el proveedor de servicios o API RESTful de forma sencilla y rápida. Dicho API se encarga de tomar los datos resultantes de la segmentación del lado del cliente para aplicar análisis sobre los mismos y almacenarlos en la base de datos.

Almacenamiento

Durante la etapa de investigación se consideró el tipo de dato que sería necesario almacenar (árboles) y debido a esto se decidió que una base de datos no relacional sería la más adecuada (MongoDB), sin embargo, durante la etapa de desarrollo se descubrió de una herramienta llamada Postgis.

Postgis es una extensión del sistema manejador de base de datos relacionales Postgresql, permite almacenar objetos GIS (Geographic Information Systems, Sistemas de información Geográfica) dentro de la base de datos y ofrece una amplia gama de funciones que permiten realizar operaciones geométricas y espaciales dentro de la misma base de datos, se consideró que podría resultar de mucha ayuda al momento de analizar los diferentes bloques rectangulares entre las diferentes segmentaciones. Por esta razón se decidió cambiar a la base de datos Postgresql.

2.2. **Antecedentes**

La segmentación de páginas Web ha sido utilizada anteriormente para ayudar en el análisis de las páginas Web. Hoy en día la segmentación de una página Web sigue siendo de gran utilidad, sin embargo, para entender el enfoque con que se realizan las segmentaciones de una página Web en la actualidad es necesario conocer cómo ha ido evolucionando a través del tiempo, es por este motivo que se incluye esta sección de antecedentes.

A continuación se presenta una recopilación de trabajos relacionados a la segmentación de páginas Web, los cuales dan contexto a la presente investigación:

Microsoft patentó VIPS en Noviembre del 2003, con el grupo de autores: Deng Cai, Shipeng Yu, Ji-Rong Wen y Wei-Ying Ma. El informe *“VIPS: a Vision-based Page Segmentation Algorithm”* (en español: VIPS: un algoritmo de segmentación de páginas basada en la visión de la misma) declara lo siguiente:

“...Este informe presenta un enfoque automático top-down, tag-free independiente para detectar estructuras de contenido web. Simula el cómo un usuario entiende el diseño de la estructura basada en su percepción visual. Comparado a otras técnicas existentes, nuestro enfoque es independiente de la representación de la

documentación subyacente tal como HTML y trabaja bien incluso cuando la estructura HTML es muy diferente de la estructura del diseño…"

Esta representa una de las primeras investigaciones en segmentación de páginas Web usando un enfoque visual, es decir, se busca hacer la segmentación basándose en la visión que tienen el usuario de la página Web, y no de la estructura como tal que forma la página.

En Diciembre del 2004 **Google** registró una patente para su método de identificar brechas visuales en una página Web. El autor fue Daniel Egnor, en su investigación *"Document segmentation based on visual gaps"* (en español: segmentación de documentos basada en brechas visuales) declara lo siguiente:

"Un documento puede ser segmentado de acuerdo a un modelo visual de dicho documento. El modelo visual está determinado de acuerdo a una cantidad de espacio blanco visible o brechas que existen dentro del documento. En una implementación, el modelo visual es usado para identificar una estructura jerárquica del documento, la cual puede ser usada posteriormente para segmentar el documento."

Con la investigación de Microsoft se dedujo que el enfoque visual es efectivo para realizar la segmentación, Google establece entonces una forma de realizar la segmentación basándose en un modelo visual y dicho modelo visual está basado a su vez en los espacios en blanco que existe entre las diferentes secciones de la página Web.

En Enero del 2008 **Yahoo** obtuvo una patente por su investigación *"Automatic Visual Segmentation of Webpages"* (en español: Segmentación visual automática de páginas web), realizado por: Deepayan Chakrabarti, Manav Ratan Mital, Swapnil Hajela y Emre Velipasaoglu. En dicha investigación se declara:

"Para proveer información valedera con respecto a una página Web, la página Web debe estar dividida en segmentos semánticamente distintos para su análisis. Un grupo de heurísticas permiten a un algoritmo de segmentación identificar un número óptimo de segmentos para una página Web dada o cualquier porción de ella con mayor precisión…"

En dicha investigación indican que la primera heurística debe estimar el número óptimo de segmentos para cualquier página Web o porción de la misma. Una segunda heurística incorpora o junta los segmentos donde el número de segmentos identificados

exceden por mucho el número óptimo recomendado, formando así un único gran segmento. Una tercera heurística incorpora o junta los segmentos correspondientes a una porción de la página con mucho espacio blanco sin usar o con muy poco contenido. Una cuarta heurística incorpora o junta segmentos de nodos que tienen un número recomendado de segmentos debajo de cierto umbral dentro de segmentos de otros nodos. Una quinta heurística analiza recursivamente y divide segmentos que corresponden a porciones de la página que sobrepasen un cierto umbral del tamaño de la porción.

Todas las investigaciones anteriormente mencionadas tenían como objetivo la segmentación de páginas Web de acuerdo a cómo un usuario las percibe visualmente, esta es la forma más aceptada en la actualidad dado que realmente es el usuario y la interpretación que éste le dé a los segmentos lo que indicará la correcta segmentación de una página Web, sin embargo, se investigaron otros enfoques, como el que se menciona a continuación.

Desde el 2012 hasta el 2014 la fundación Open Preservation Foundation y The SCAPE Project soportan la investigación y desarrollo del "Pagelyzer", cuyos responsables son: Stéphane Gançarski y Matthieu Cord. Y sus contribuidores: Andrés Sanoja, Carl Wilson, Marc Law y Zeynep Pehlivan.

El **Pagelyzer** es una herramienta usada para comparar dos versiones de una misma página Web y averiguar si son similares o no, la herramienta se apoya en varias de investigaciones de segmentación de páginas Web como por ejemplo en el algoritmo de VIPS y el BoM (ambos mencionados anteriormente).

Basado en "*A Web Page Segmentation Algorithm"* (en español: Un Algoritmo para la Segmentación de Páginas Web), posee una combinación de métodos de comparación tanto estructurales como visuales, incrustados en un modelo estadístico discriminatorio de "*a visual similarity measure designed for Web pages that improves change detection"* (en español: una medida de similitud visual diseñada para páginas Web que mejoran la detección de cambio), el cual permite llevar el seguimiento de los cambios, todo adaptado para el archivamiento Web.

En la figura 1 se puede observar un ejemplo del funcionamiento del Pagelyzer, las zonas marcadas en verde es donde se produjo un cambio con respecto a la otra versión, y la zona marcadas en rojo se mantuvieron iguales.

Figura 1[3]: Ejemplo de la herramienta Pagelyzer.

A continuación se presentan una serie de trabajos los cuales representan las bases para el desarrollo del presente Trabajo Especial de Grado.

Andrés Sanoja y Stéphane Gançarski. *"Yet Another Hybrid Segmentation Tool"*. Poster y artículo corto presentado en la conferencia iPRES 2012 en la Facultad de Información de la Universidad de Toronto, Canadá en el 2012.

En este trabajo se presenta un prototipo desarrollado en el contexto de archivos Web (comparación de páginas, rastreo y extracción de información). Analiza las páginas basándose en la información del árbol DOM de la misma y en su representación visual. Esta herramienta implementa una versión modificada del VIPS (Vision-based Page Segmentation) con el objetivo de mejorar la precisión de la extracción del bloque visual y la construcción jerárquica. La herramienta aquí presentada sirvió como primer prototipo y prueba de concepto de lo que posteriormente se consolidaría en la herramienta Block-o-Matic (BoM).

Andrés Sanoja y Stéphane Gançarski. *"Block-o-Matic: a Web Page Segmentation Tool and its Evaluation"*. Presentado en la conferencia de BDA (Big Data Analytics) 2013, en Nantes, Francia.

[3] Imagen recuperada de Pagelyzer en SCAPE project, http://scape-project.eu/leaflets/monitor-your-web-content-with-pagelyzer

En su informe presentan un prototipo de segmentación de páginas Web llamado “Block-o-matic” y una contraparte llamada “Block-o-manual”, para la segmentación manual. La principal idea del “Block-o-manual” es evaluar la correctitud del algoritmo de segmentación. Se propone la construcción de una base de datos construida para almacenar las evaluaciones realizadas que se consideran “correctos” (ground truth). En este trabajo se evidencia la necesidad de evaluar la segmentación y, al mismo tiempo, de contar con una herramienta de propósito general que soporte el proceso. Tomando en consideración que no existe ninguna alternativa en la comunidad de software ni académica al momento de la publicación del trabajo.

Andrés Sanoja y Stéphane Gançarski. *“Block-o-matic: A web page segmentation framework”*. Publicado en el Multimedia Computing and Systems (ICMCS), 2014 International Conference por IEEE en el 2014.

En su informe describen el Block-o-Matic. Es un framework de segmentación de páginas Web. Se basa en un enfoque híbrido inspirado en los métodos de procesamiento de documentos (document processing) automatizados (*e.g.* documentos escaneados, OCR) y en las técnicas de segmentación de contenido basados en pistas visuales y el DOM. El proceso de segmentación está dividido en tres fases: análisis, entendimiento y reconstrucción de la página Web.

Un proceso de evaluación es propuesto con el fin de realizar una evaluación basada en un *ground truth* de 400 páginas clasificadas en 16 categorías. Los resultados arrojados por Block-Matic aseguran ser prometedores. Este trabajo consolida los conocimientos sobre BoM y su primera versión como herramienta de propósito general. Presenta un método de evaluación el cual evidencia la necesidad y justificación de la construcción de una herramienta para la segmentación manual.

Andrés Sanoja y Stéphane Gançarski. *“Web page segmentation evaluation”.* Presentado en la 30º reunión anual del ACM Symposium on Applied Computing en el 2015.

En su informe presentan un framework para evaluar algoritmos de segmentación de páginas Web. Definen un modelo de evaluación que incluye diferentes métricas para evaluar la calidad de la segmentación obtenida dado un algoritmo. Dichas métricas calculan la distancia entre la segmentación obtenida y una segmentación construida manualmente, que sirve como *ground truth*. Aplican el framework a cuatro algoritmos considerados estados-del-arte (BoM, Block Fusion, VIPS y JVIPS) en diferentes categorías (tipos) de páginas Web. Los resultados muestran que los algoritmos

probados usualmente funcionan muy bien para la extracción de texto, pero pueden llegar a tener serios problemas para la extracción geométrica. También muestran que la calidad relativa de un algoritmo de segmentación depende de la categoría de la página segmentada.

2.3. **Trabajos de Referencia**

Este trabajo se enmarca en la evaluación de la segmentación de páginas Web. Es la continuación de un trabajo futuro planteado en el trabajo realizado por Andrés Sanoja y Stéphane Gançarski publicado en Junio del 2016: *Block-based Migration from HTML4 Standard to HTML5 Standard in the Context of Web Archives*, o en español: Migraciones Basadas en Bloques del Estándar HTML4 al Estándar HTML5 en el Contexto de Archivos Web. La orientación de esta recopilación es mostrar la importancia de producir una herramienta de segmentación manual, orientada a los usuarios..

En la investigación "*Block-based Migration from HTML4 Standard to HTML5 Standard in the Context of Web Archives",* se presentan dos herramientas mediante las cuales es posible realizar el proceso de migración de formato, estas son: BoM y MoB (Manual-design-Of-Blocks), BoM representa el algoritmo de segmentación el cual segmenta una página Web de forma automática, y el MoB es la herramienta de segmentación manual.

En el presente trabajo investigativo se busca la creación de una herramienta similar al MoB, una herramienta capaz de permitirle al usuario la segmentación de una página web, arrojando como resultado un documento donde se represente una estructura de árbol de segmentación basado en el árbol DOM de la página segmentada, junto al conjunto de rectángulos visuales que conforman la segmentación visual, pudiendo almacenar dicho resultado en un repositorio de datos para su posterior análisis.

La herramienta MoB se encuentra solo como una propuesta, como un prototipo de baja fidelidad. Es por esa razón que este trabajo representa una evolución de la herramienta MoB, la cual fue concebida por la necesidad de poder asegurar un *ground truth* para apoyar los algoritmos de segmentación. En las secciones siguientes se describen las dos herramientas involucradas: BoM y MoB.

2.3.1. BoM

Block-o-Matic (BoM), es una herramienta que da solución a la segmentación automatizada de una página Web. La herramienta implementa un algoritmo de segmentación que hace posible la segmentación automatizada de la página. Uno de los aspectos claves de esta herramienta es que no requiere conocimientos previos del contenido de la página a segmentar. La segmentación es guiada únicamente por las reglas heurísticas definidas por el estándar del W3C Web (*W3C Recommendation-HTML5* , 2014).

Debido a la lógica de segmentación usada por BoM para realizar la segmentación, permite (en teoría), segmentar cualquier tipo de página Web, ya que, detecta los bloques usando las categorías de contenido de HTML5 en vez de hacer uso de los nombres de las etiquetas o cualidades relevantes de los textos.

La investigación se apoya en el hecho de que una página web está asociada a tres estructuras: un árbol DOM, una estructura de contenido, y una estructura lógica. El árbol DOM representa los elementos HTML de la página resultado del rendering hecho por un navegador. La estructura geométrica organiza el contenido basado en una categoría y su geometría (en rectángulos). La estructura lógica es el resultado de la correspondencia existente entre la estructura del contenido y la percepción significativa que le da el usuario a los bloques mediante reglas heurísticas. El proceso de segmentación está dividido en tres fases: el análisis, el entendimiento y la reconstrucción de la página web. En la fase de análisis se descompone la página, identificando elementos del DOM candidatos a ser bloques. En la fase de entendimiento se identifican los bloques compuestos (*composite*) y se combinan para formar segmentos o bloques coherentes o significativos. Se forman bloques cuyo tamaño relativo no sea mayor que un parámetro de granularidad. En la fase de reconstrucción se organizan los segmentos en un grafo (árbol general) y se le adjuntan los sub-árboles del DOM correspondientes a cada bloque. Esta estructura sirve como descriptor de la segmentación, y es su resultado.

En la figura 2 se puede observar el proceso que realiza el algoritmo desde que se obtiene el documento de la página web hasta que se obtiene el archivo resultado con una sintaxis parecida a un lenguaje de marcado donde se representa el árbol de la página segmentado.

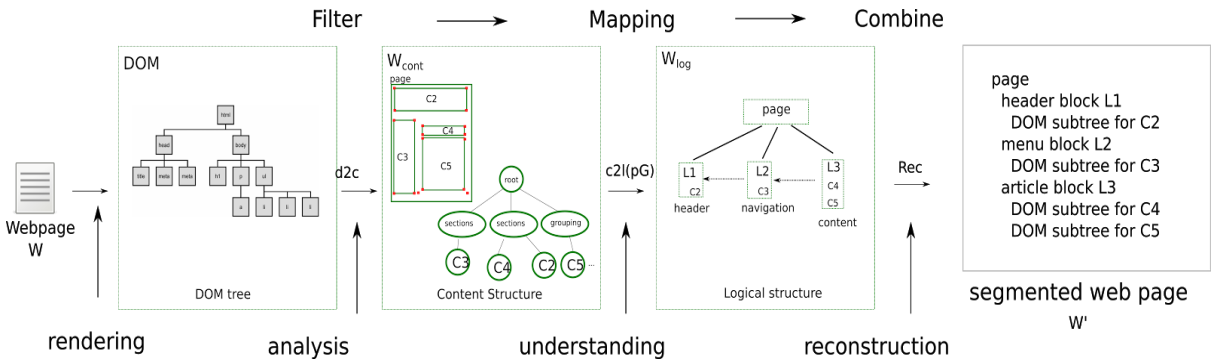

Figura 2[4]: Proceso de segmentación realizada por BoM.

En la figura 3 se puede observar un ejemplo de la herramienta BoM en acción, donde se ha aplicado una segmentación a una página Web. Se ha utilizado una granularidad de 0.5 (siendo 0 bloques pequeños y 1 el tamaño relativo de la página Web completa). La segmentación es representada usando rectángulos de diferentes colores. Cada rectángulo tiene la finalidad de representar un segmento de la página. Se utilizan diferentes colores para diferenciar los tipos de los segmentos: el azul sería un segmento del tipo “contenido”, el verde seria un segmento del tipo “contenedor” y el morado es un segmento del tipo “predefinido”.

En este TEG se tomará en cuenta esta diferenciación de colores para poder indicarle visualmente al usuario el tipo de segmento que se está segmentando, aunque esto puede ser modificado por el usuario si no está de acuerdo con dicha clasificación del segmento.

[4] Imagen recuperada de Block-o-Matic, http://bom.ciens.ucv.ve/

Figura 3[5]*: Segmentación automática realizada por BoM.*

2.3.2. MoB

Manual-design-Of-Blocks (MoB), es un prototipo utilizado para establecer el *ground truth* acerca de la segmentación de una página Web.

Dicha herramienta fue concebida como una extensión del explorador Chrome para que usuarios expertos pudiesen realizar la segmentación. Los usuarios crean bloques dependiendo de los elementos Web y sus respectivas jerarquías. Se obtiene un grafo de bloques (dadas las jerarquías) o simplemente una segmentación plana (solo bloques terminales). Ambas segmentaciones producen un documento XML el cual representa el árbol de segmentación basado en el árbol DOM de la página segmentada. Produce también un conjunto de rectángulos presentados de manera visual. Estos resultados son almacenados en un repositorio de datos para su posterior evaluación.

[5] Imagen hace referencia de la herramienta BoM en la página: http://www.appropedia.org/MY4777

A continuación se presenta una serie de figuras con ejemplos del funcionamiento del prototipo MoB.

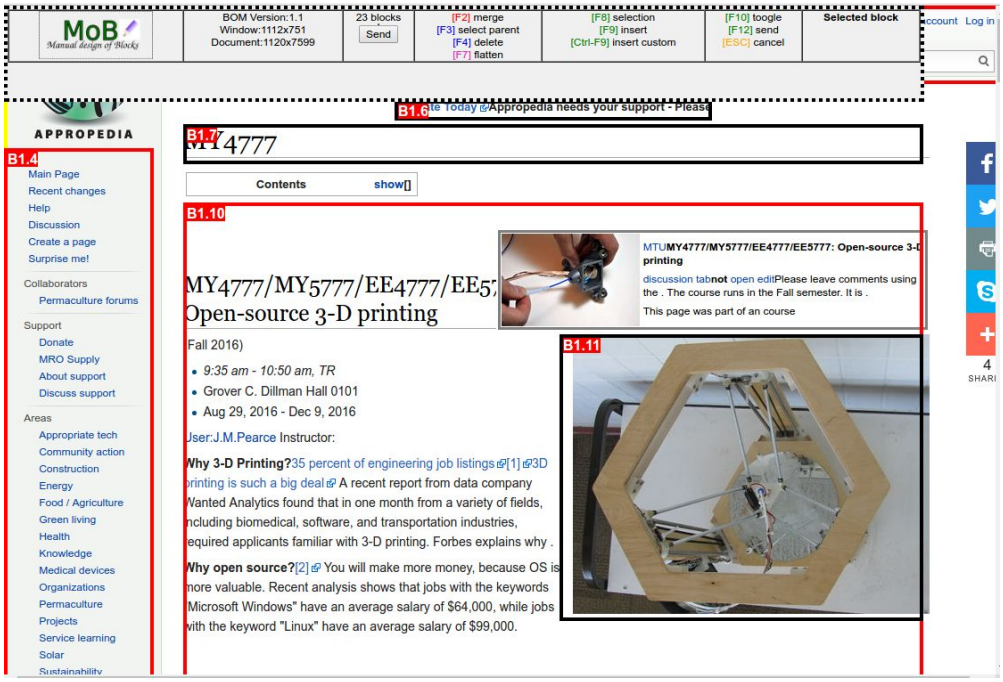

Figura 4[6]: Segmentación previa realizada por BoM.

En la figura 4 se evidencia el hecho de que MoB posee un panel en la parte superior donde muestra la leyenda de los comandos a usar para realizar la segmentación manual. Se le presenta al usuario una segmentación realizada por BoM. Se le propone al usuario aceptar o modificar la segmentación propuesta.

Si el usuario desea modificar la segmentación, por ejemplo agregar un bloque, debe presionar F9 y hacer click en el elemento que desea segmentar. MoB mostrará un mensaje con la lista de posibles elementos que se encuentra debajo del click. Este proceso es reflejado en la figura 5.

[6] Todas las imágenes hacen referencia a la herramienta MoB en la página: http://www.appropedia.org/MY4777

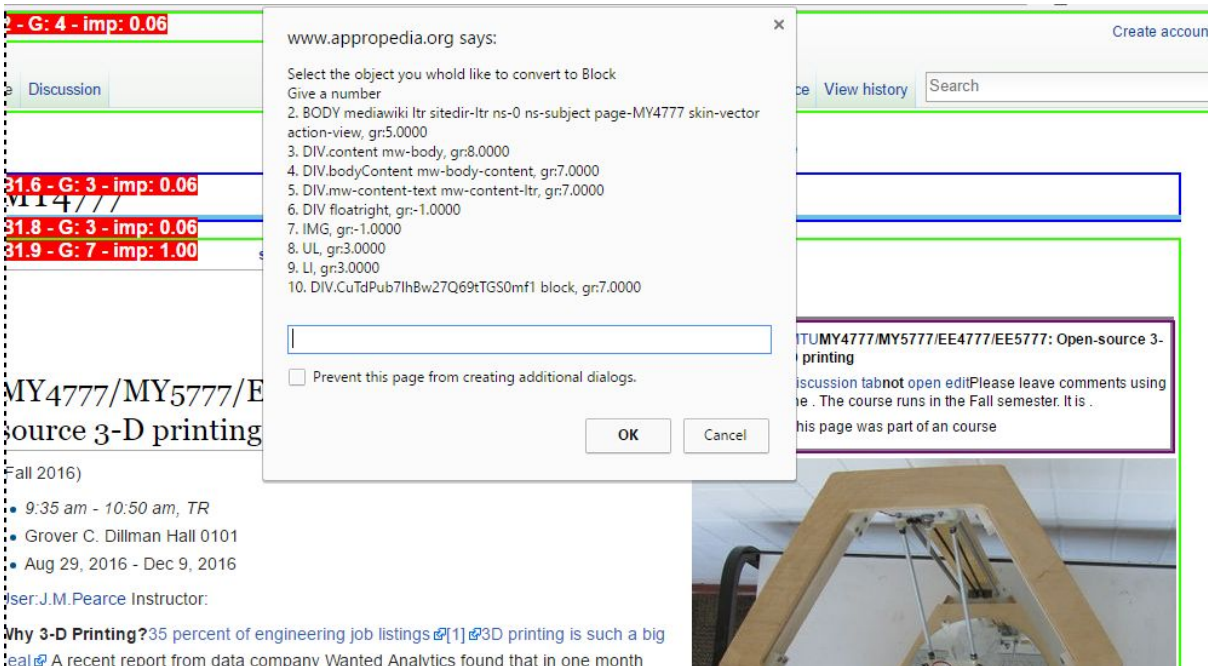

Figura 5: Cuadro de diálogo.

Se debe introducir el número del elemento y presionar el botón de “ok”, una vez hecho esto se resaltará dicho elemento indicando que se ha segmentado. Obteniendo un resultado como se muestra en la figura 6.

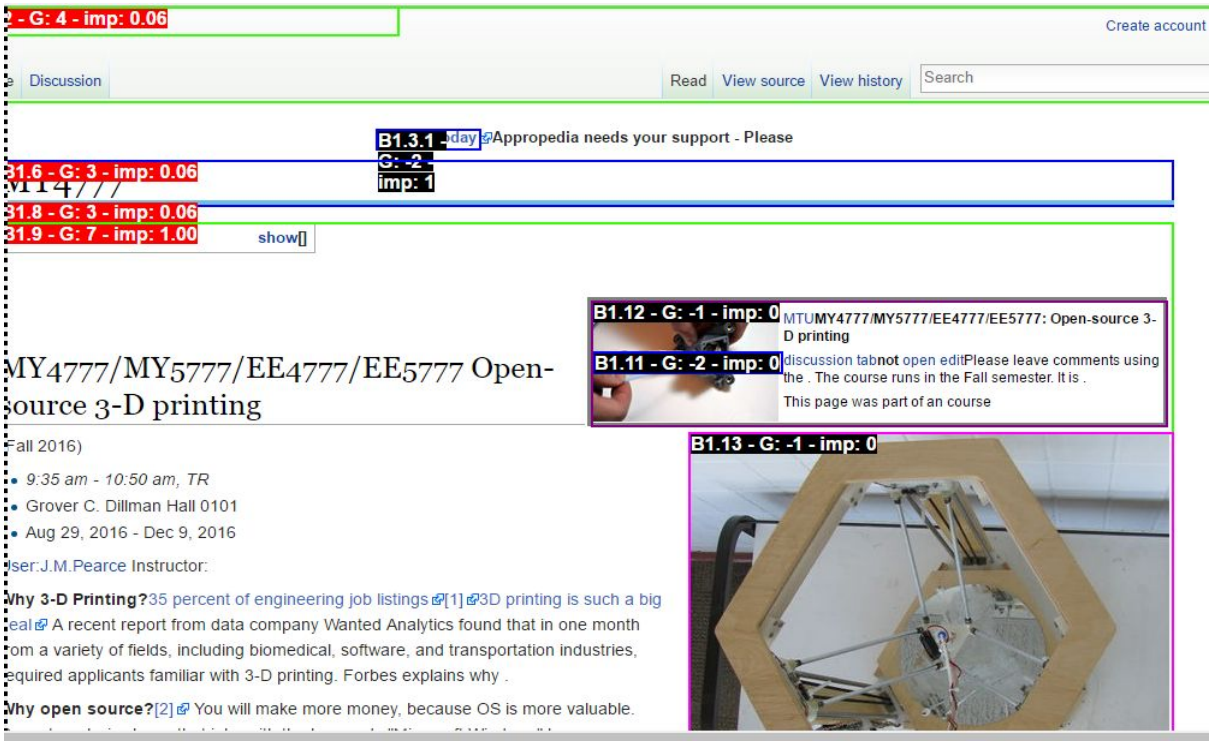

Figura 6: Bloque segmentado.

Este proceso resulta tedioso, complicado y propenso a errores, incluso para usuarios expertos, por esta razón el presente trabajo de investigación busca mejorar su usabilidad. Por ejemplo, resaltar tentativamente el elemento que se desea segmentar permitiéndole al usuario una retroalimentación activa, y obtener una vista previa de acción antes de realizarla. Se desea crear también un sistema de puntajes, el cual motive a los usuarios a realizar segmentaciones.

Aun cuando esto es solo un ejemplo de las funcionalidades requeridas para la nueva versión de MoB, forma parte del TEG explorar nuevos requerimientos que amplíen la aplicación desde el punto de vista funcional, no funcional y de usabilidad.

2.3.3. Proyecto SCAPE

Scalable Preservation Environments ó Ambientes de Preservación Escalables, el proyecto SCAPE se creó en el 2011 co-fundado por la Unión Europea, fue cerrado en el 2014, durante su época de actividad su misión consistía en desarrollar servicios escalables, los cuales, a través de una plataforma de código abierto se encargaban del manejo, supervisión y orquestación de procesos semi-automáticos para colecciones de objetos digitales complejos, heterogéneos y a gran escala.

BoM y MoB son productos de este proyecto. Forman parte de la aplicación Pagelyzer, una herramienta usada para el control de versiones de una página Web, creada en el 2012 por Stéphane Gançarski y Matthieu Cord y en la que contribuyeron Andrés Sanoja, Carl Wilson, Marc Law y Zeynep Pehlivan. En la sección 2.4 se profundiza en la utilidad y funcionamiento de esta herramienta.

2.4. Trabajos Relacionados

Como se ha mencionado anteriormente, en la actualidad la necesidad de la segmentación de páginas Web sigue estando presente, por esa razón se ha mantenido el desarrollo de herramientas como FitLayout.

FitLayout es un framework extensible de segmentación de páginas Web y análisis, creado en el 2014 por Radek Burget. Define una API de Java genérica para representar una página Web y sus divisiones de áreas visuales, provee una base para la implementación de algoritmos de segmentación de páginas con una interfaz de aplicación común. El framework incluye herramientas que permiten el procesamiento posterior al resultado de la segmentación a través de diferentes métodos de clasificación de texto o visual. Una captura del framework se puede observar en la figura 7. Al igual que el sistema que se desarrolla en este trabajo, permite observar los resultados de las diferentes segmentaciones realizadas por diferentes algoritmos.

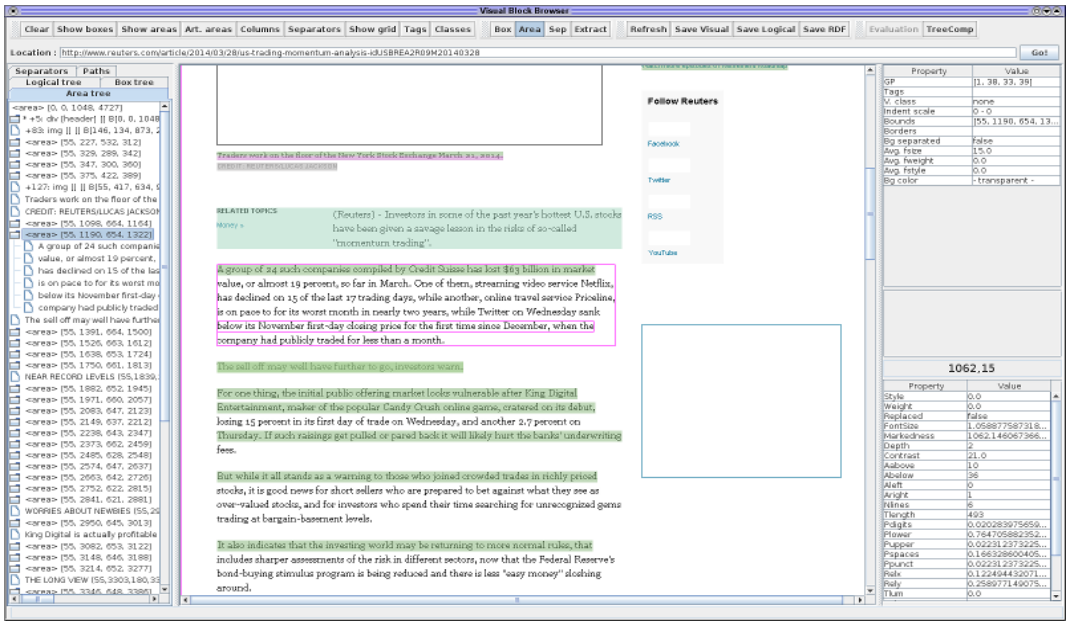

Figura 7[7]: Ejemplo de la herramienta FitLayout.

Chrome DevTools es una herramienta desarrollada por el equipo de Google la cual provee una serie de funcionalidades que le permiten a los desarrolladores o a cualquier usuario la habilidad de poder interactuar con los elementos del DOM de alguna página Web a tiempo real. Entre sus funcionalidades está:

- Inspecciona y edita sobre la marcha cualquier elemento del árbol del DOM en el panel Elements.
- Visualiza y cambia las reglas de CSS que se aplican a cualquier elemento seleccionado en el subpanel Styles.
- Visualiza los cambios realizados en tu página localmente en el panel Sources.

[7] Imagen recuperada de FITLayout, http://www.fit.vutbr.cz/~burgetr/FITLayout/

Un ejemplo de la herramienta en acción se puede vislumbrar en la figura 8, donde se observa la herramienta iluminando un elemento DOM de una página Web cualquiera y presentando la información de dicho elemento para su posible edición.

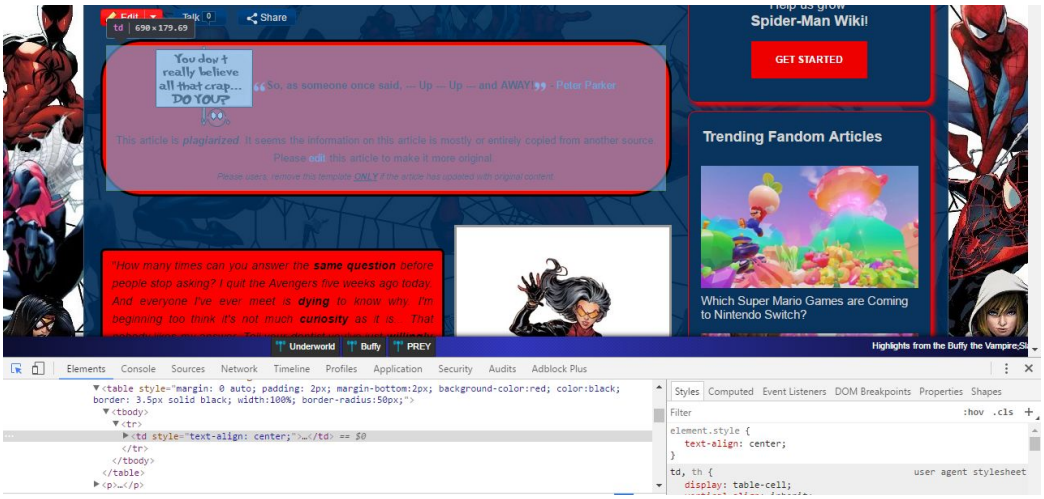

Figura 8[8]*: Ejemplo de la herramienta Chrome DevTools.*

Es importante destacar esta herramienta ya que parte de su funcionamiento (el poder iluminar los elementos del DOM que se busca editar) es muy similar a la forma en el que el usuario escoge los elementos del DOM a segmentar en la extensión del presente trabajo.

[8] Imagen tomada por el autor, el sitio usado de ejemplo fue: http://spiderman.wikia.com/wiki/Jessica_Drew_(Earth-616)

CAPÍTULO III: MARCO METODOLÓGICO

En este capítulo se describen las metodologías y métodos de desarrollo a utilizar durante el TEG. En la sección 3.1 se describe la metodología de desarrollo Kanban. En la sección 3.2 se describe la metodología para la integración de componentes.

3.1. Metodología de Trabajo Kanban

Kanban es un método para gestionar el trabajo intelectual, con énfasis en la entrega justo a tiempo, mientras no se sobrecarguen los miembros del equipo. En este enfoque, el proceso, desde la definición de una tarea hasta su entrega al cliente, se muestra para que los participantes lo vean y los miembros del equipo tomen el trabajo de una cola.

Básicamente se separa el flujo del desarrollo en etapas diferenciales e incrementales, se utilizan tarjetas para representar las actividades o elementos dentro de dichas etapas, indicando así el recorrido de un elemento entre las etapas o las actividades a desarrollar en dichas etapas; esto permite la organización de las actividades y el “camino a seguir” para realizarlas, permitiendo que el individuo no se sobrecargue con actividades y tenga una visión clara de dónde está y hacia dónde va. Para la aplicación de esta metodología se usa una herramienta llamada Trello.

Trello es un software de administración de proyectos con interfaz Web. Emplea el sistema kanban, lleva un registro de actividades a través de tarjetas virtuales las cuales organizan las diferentes tareas que se deben realizar, permite agregar listas, adjuntar archivos, etiquetar eventos, agregar comentarios y compartir tableros.

3.2. **Desarrollo Adaptable de Software**

Para la integración de los diferentes componentes se usa la metodología de ASD (Adaptive Software Development o en español: Desarrollo Adaptable de Software) fue desarrollada por Jim Highsmith y Sam Bayer a comienzo del año 1990. Esta metodología es guiada por el principio de que el desarrollo del software debe adaptarse a los cambios que puedan ocurrir, en lugar de luchar contra ellos.

Sus principales características son:

- Es iterativo.
- Orientado a los componentes del software (las funcionalidades y características que el producto va a tener) más que las tareas en las que se alcanzaran dicho objetivo.
- Es tolerante a los cambios.
- La revisión de los componentes permite aprender de los errores y volver a iniciar el ciclo de desarrollo.

En la mayoría de las metodologías de desarrollo se tiene un ciclo de vida como: Planificación-Diseño-Construcción. Sin embargo, ASD posee un ciclo iterativo no lineal, donde cada ciclo puede iterar y ser modificado al tiempo que otro es ejecutado.

El ASD utiliza un ciclo de desarrollo dinámico conocido como: Especular-Colaborar-Aprender. La representación del ciclo es mostrada en la figura 9 que se presenta a continuación.

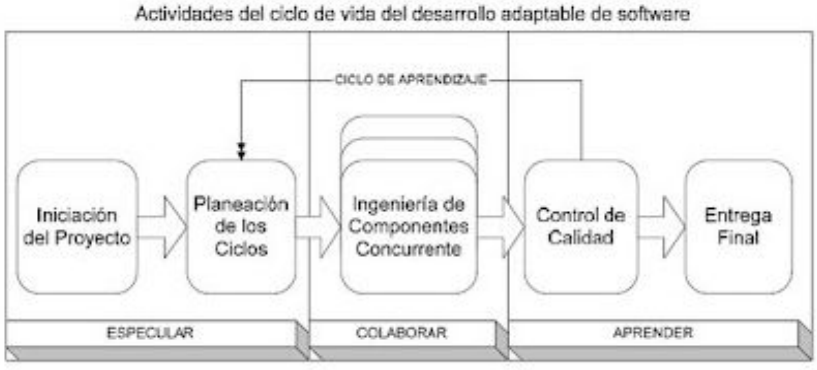

Figura 9[9]: Ciclo de la metodología de desarrollo ASD.

[9] Imagen recuperada de ASD (Adaptive Software Development), http://adaptivesoftwaredevelopment.blogspot.com/

En la etapa de **especulación** se busca establecer los principales objetivos y metas del proyecto y comprender las limitaciones (zonas de riesgo) del mismo. En ASD se realizan estimaciones de tiempo sabiendo que pueden sufrir desviaciones. Se decide el número de iteraciones para consumir el proyecto, prestando atención a las características que pueden ser utilizadas por el cliente al final de la iteración. Estos pasos se puede volver a examinar varias veces antes de que el equipo y los clientes están satisfechos con el resultado.

En la etapa de **colaboración** se centra la mayor parte del desarrollo manteniendo una componente cíclica. Es importante establecer una coordinación que asegure que lo aprendido por un equipo se transmite al resto y no tenga que volver a ser aprendido por los otros equipos.

En la etapa de **aprendizaje** se centra en capturar lo que se ha aprendido, tanto positivo como negativo mediante retroalimentaciones o reuniones de grupo, esto ayuda a soportar y solucionar de una mejor manera el constante cambio que puede tener el proyecto y su adaptación.

Ventajas

- La tercera fase del ciclo de vida, revisión de los componentes, sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo.
- Apunta hacia el Rapid Application Development (RAD), el cual enfatiza velocidad de desarrollo para crear un producto de alta calidad, bajo mantenimiento involucrando al usuario lo más posible.
- Utiliza información disponible acerca de cambios para mejorar el comportamiento del software.
- Promulga colaboración, la interacción de personas.
- Anticipa cambios y trata automáticamente con ellos dentro de un programa en ejecución, sin la necesidad de un programador.

Desventajas

- Aunque el ciclo entre el aprendizaje y la especulación permite entregar productos con alta calidad, la prolongación de dicho ciclo por errores o cambios que no son detectados en reuniones anteriores afecta tanto a la calidad del producto como a su costo total.

- Dado a que es una metodología ágil implica no realizar procesos que son requeridos en las metodologías tradicionales o por lo menos no realizarlos en procesos diferentes, lo cual implica que empresas grandes las cuales necesitan llevar un mayor control a procesos y personas, tener tareas asignadas a un estado o proceso específico, y en las cuales dicho incremento de procesos no afectan en gran medida al costo final del producto, para dichas empresas el elegir una metodología tradicional resulta mucho más rentable tanto por el gran volumen de personal, de productos, y de costos que se manejan y para los cuales se tendrá un mayor control.

Se busca emplear esta metodología de desarrollo debido a que la herramienta a construir forma parte de un gran sistema de herramientas que se relacionan entre sí y son dependientes entre sí (*i.e* los resultados obtenidos de una son los datos de entrada requeridos de otras), es por esto que se pueden presentar muchos cambios a lo largo del proyecto y se debe ir aprendiendo a medida que se va desarrollando e ir propagando dicho aprendizaje a todos los equipos de desarrollo.

CAPÍTULO IV: DESARROLLO DE LA SOLUCIÓN

4.1. Descripción general de la solución

En el presente Trabajo Especial de Grado se presenta el desarrollo de una herramienta utilizando tecnologías Web que permiten a un usuario segmentar manualmente una página Web para conformar una base de datos (*ground truth*). Esta herramienta transforma la segmentación hecha por el usuario en un árbol de segmentación basado en el árbol DOM de la página segmentada y un árbol estructural de bloques asociados al diseño de la página. Los datos son almacenados en una base de datos para su posterior recuperación y análisis. Se repite el mismo proceso para diferentes usuarios. Los resultados de la segmentación, hechos por diferentes usuarios, de una misma página Web son analizados para encontrar un promedio entre ellos. Mientras mayor sea el número de resultados mayor será la probabilidad de que el análisis de los mismos arroje lo que sería la mejor segmentación de dicha página Web.

La mejor segmentación puede ser comparada entonces con la segmentación resultante de un algoritmo de segmentación (*e.g* el BoM) para evaluar su correcto funcionamiento. Se utiliza un sistema de puntajes para alentar a los usuarios a realizar segmentaciones, además de que las mismas puntuaciones obtenidas pueden ser usadas para medir la calidad de la segmentación. La descripción realizada anteriormente puede visualizarse a través de la figura 10, las líneas en rojo delimitan los límites de este TEG.

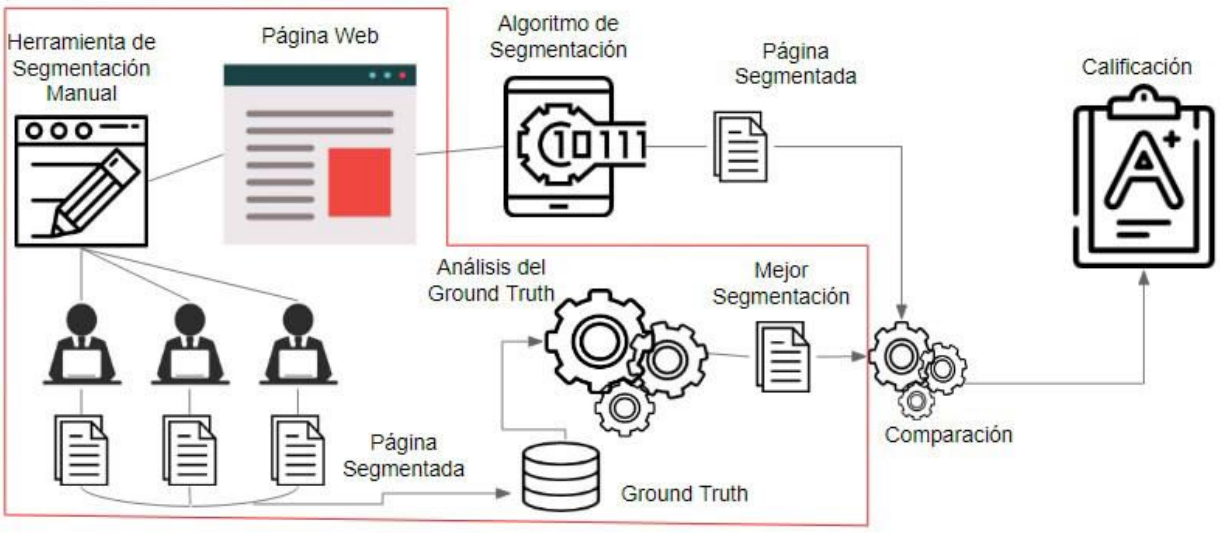

Figura 10: Modelo de la propuesta de TEG.

4.2. Arquitectura de la solución

Para aclarar la arquitectura a utilizar, se presenta el modelo de la arquitectura MVC en la figura 11, y después se hace una descripción del comportamiento y las interacciones entre cada uno de los componentes.

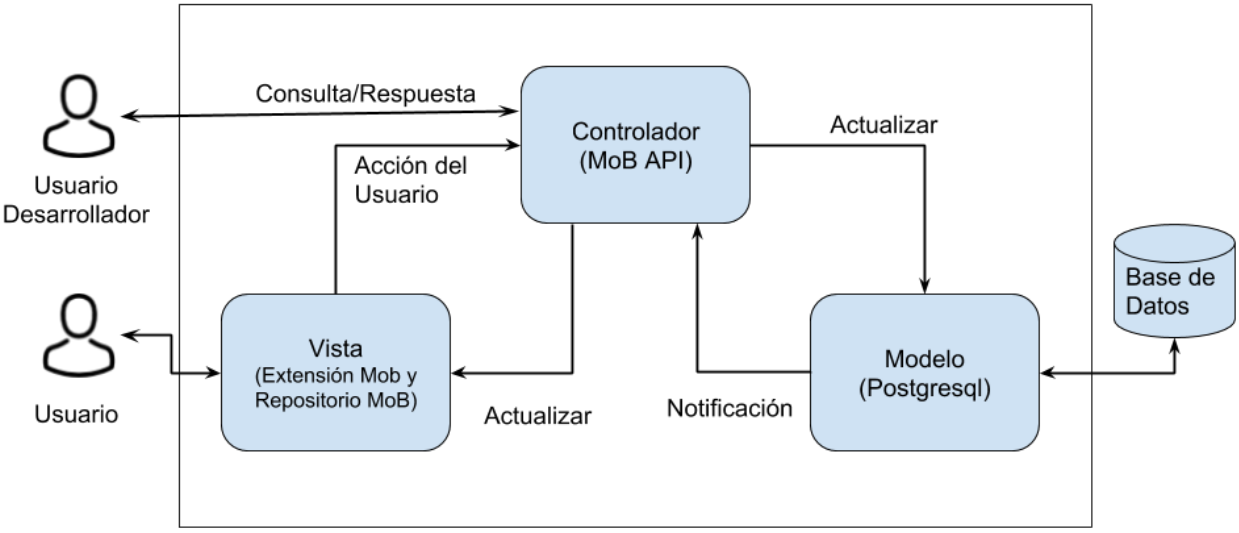

Figura 11*: Modelo de la Arquitectura MVC del Sistema.*

Vista:

Está comprendida por la extensión de explorador Chrome (Extensión MoB) y el sitio Web que funciona como repositorio de datos (Repositorio MoB), ambas constan de interfaces desarrolladas con HTML5, CSS3 y JS (*c.f* Capítulo 2.1.8.1). El usuario interactúa con las interfaces de la vista, desencadenando diferentes acciones (consultar información, enviar segmentación, inicio de sesión, entre otras), estas acciones son enviadas por la vista hasta el controlador.

Controlador:

Está comprendido por la API RESTful de MoB (MoB API), la cual está desarrollada con Python 3 y Flask (*c.f* Capítulo 2.1.8.3). La API ofrece servicios que pueden ser consultados directamente por un usuario desarrollador, también maneja las acciones del usuario enviadas desde la vista, dependiendo de la acción actualiza o consulta información en los modelos de datos (modelo) antes de enviar su respuesta.

Modelo:

El modelo está comprendido por los modelos de la base de datos, manejada por el manejador de base de datos Postgresql y su componente Postgis (*c.f* Capítulo 2.1.8.4). El controlador actualiza los datos existentes en los modelos de datos o realiza consultas sobre estos, el modelo se conecta con la base de datos para devolver entonces notificaciones dependiendo de las acciones realizadas.

4.3. Ciclos de Desarrollo

Los ciclos de desarrollo comprenden los ciclos realizados para desarrollar las funcionalidades del sistema. Cada uno consta de tres etapas: especulación, colaboración y aprendizaje (*c.f* Capítulo 3.2). Cada uno estuvo planificado para un periodo de tiempo de un mes (el cual puede variar), y a la vez, en cada ciclo pueden ocurrir varias iteraciones hasta desarrollar exitosamente todos los requerimientos comprendidos en el ciclo, cabe acotar que los ciclos pueden llegar a desarrollarse en paralelo en alguna de sus iteraciones. A continuación, en la sección 4.3.1 se presenta el ciclo de desarrollo de la extensión MoB, en la sección 4.3.2 se describe el ciclo de desarrollo de la API RESTful de MoB (*c.f* Capítulo 2.1.6), en la sección 4.3.3 se

presenta el ciclo de desarrollo del Repositorio de MoB, en la sección 4.3.4 se describe el ciclo donde se desarrolla la estrategia para abordar la mejor segmentación, en la sección 4.3.5 se presenta un ciclo realizado para aplicar ciertas mejoras al sistema en general. Al final de cada ciclo se indican los objetivos logrados con el desarrollo del mismo.

4.3.1. 1er Ciclo: Extensión MoB

Este ciclo consiste en el desarrollo de la extensión de Chrome denominada MoB, la cual permitirá a los usuarios: registrarse en el sistema de MoB, iniciar sesión, editar sus datos de usuario, iniciar la herramienta de segmentación manual de páginas Web, enviar todos los datos a la API de MoB, entre otros.

- **Especulación:**

Mediante los procesos de especulación realizados en el desarrollo de este ciclo, se pudo establecer los requerimientos que debía satisfacer la extensión con la finalidad de poder obtener todos los datos necesarios para el posterior procesamiento y almacenamiento de las páginas Web y sus respectivas segmentaciones.

A continuación en la figura 12 y 13 se presentan los diagramas de Casos de Uso seguido por la tabla 1 hasta la tabla 6, las cuales contienen la descripción de cada uno de los casos de uso. Cabe acotar que los casos de usos derivados del UC.6 no fueron descritos en tablas pues más adelante se hace una descripción de cada uno de ellos.

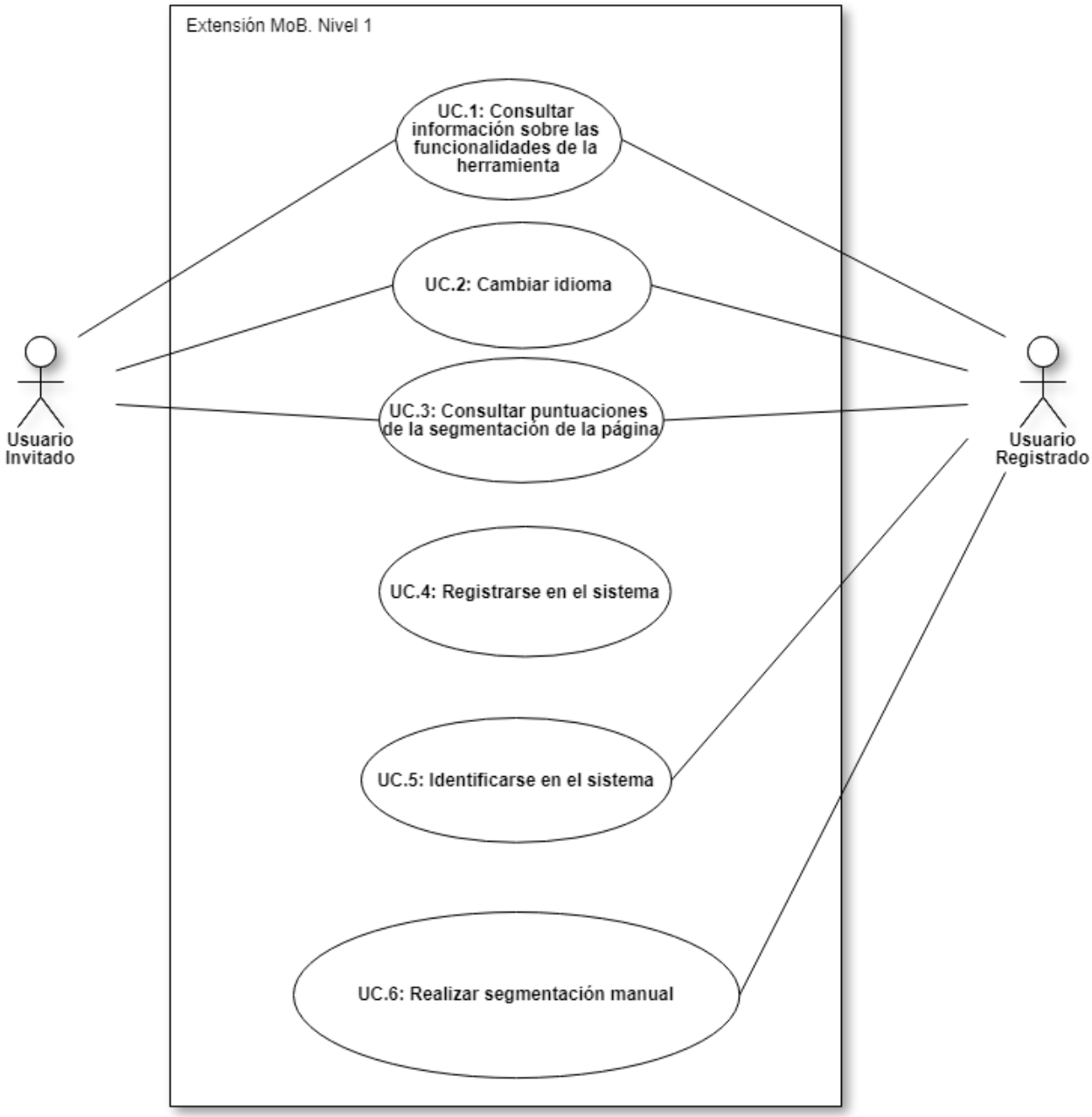

Figura 12: Diagrama de Casos de Uso de la Extensión MoB, Nivel 1

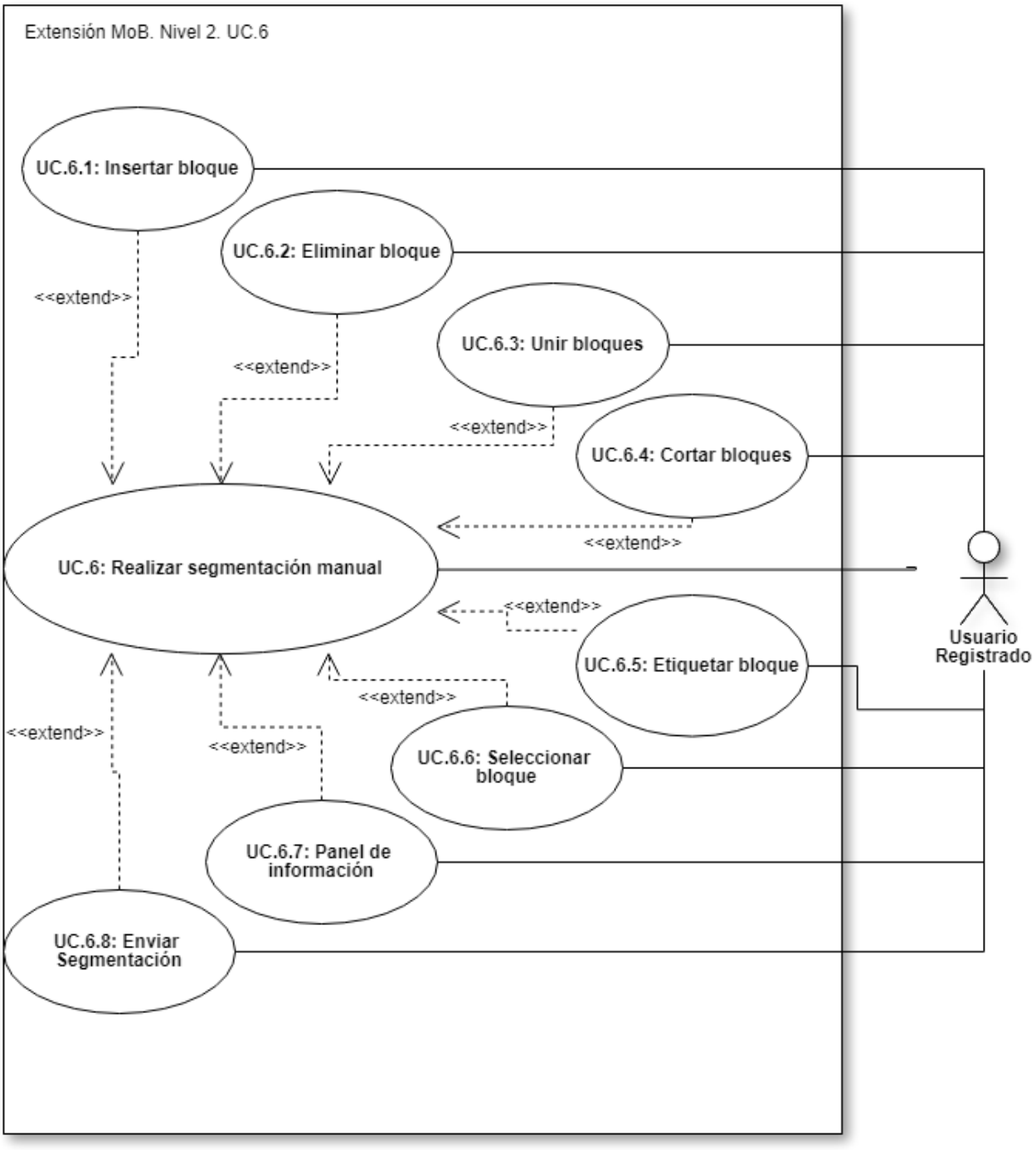

Figura 13: Diagrama de Casos de Uso 6 de la Extensión MoB, Nivel 2

Nombre	**UC.1:** Consultar información sobre las funcionalidades de la herramienta
Descripción	El usuario puede consultar la información sobre las

	diferentes acciones que se pueden realizar con la herramienta de segmentación manual MoB.
Actor	Usuario Invitado y Usuario Registrado
Flujo de Eventos	
Flujo Básico	
El caso de uso inicia cuando el usuario selecciona el icono de información (i). 1. El sistema cambia de pantalla y presenta un cuadro con varios iconos, cada uno representando una acción de la herramienta de segmentación. 2. El usuario selecciona alguno de los iconos. 3. El sistema presenta la información asociada al icono seleccionado por el usuario. 4. El caso de uso termina	
Flujo Alternativo	
Título	**Descripción**
N/A	N/A
Precondiciones	
Título	**Descripción**
N/A	N/A
Postcondiciones	
Título	**Descripción**
Éxito	El usuario obtiene información sobre las acciones pueden ser realizadas con la herramienta de segmentación manual MoB.
Fracaso	N/A

Tabla 1: Extensión MoB UC.1

Nombre	**UC.2:** Cambiar idioma
Descripción	El usuario puede cambiar el idioma de la extensión entre español, inglés o francés.
Actor	Usuario Invitado y Usuario Registrado
Flujo de Eventos	
Flujo Básico	

El caso de uso inicia cuando se activa el icono de la extensión. 1. El sistema presenta los tres posibles idiomas: español, inglés o francés. 2. El usuario selecciona uno de los tres idiomas ofrecidos. 3. El lenguaje del sistema cambia de acuerdo a la selección. 4. El caso de uso termina.	
Flujo Alternativo	
Título	**Descripción**
N/A	N/A
Precondiciones	
Título	**Descripción**
N/A	N/A
Postcondiciones	
Título	**Descripción**
Éxito	El lenguaje de la extensión se modifica según el lenguaje seleccionado.
Fracaso	El lenguaje no se modifica, se queda en inglés por defecto.

Tabla 2: Extensión MoB UC.2

Nombre	**UC.3:** Consultar puntuaciones de la segmentación de la página
Descripción	El usuario puede observar las mayores puntuaciones de las segmentaciones en sus diferentes granularidades de una página Web.
Actor	Usuario Registrado
Flujo de Eventos	
Flujo Básico	
El caso de uso inicia cuando el usuario selecciona el icono de puntuaciones (un trofeo). 1. El sistema cambia de pantalla y presenta una tabla con dos pestañas. 2. El usuario selecciona alguna de las pestañas (puntaje individual o puntaje global).	

3. El sistema presenta los puntajes de las segmentaciones en las 10 granularidades de la pestaña seleccionada por el usuario. 4. El caso de uso termina.	
Flujo Alternativo	
Título	**Descripción**
Usuario no identificado	1. El sistema cambia de pantalla y presenta un aviso de que debe identificarse para poder observar las puntuaciones.
Página Web no almacenada	1. El sistema presenta un aviso de error donde indica que dicha página Web no posee segmentaciones dentro del sistema.
Precondiciones	
Título	**Descripción**
Identificarse	El usuario debe identificarse en el sistema para poder observar las puntuaciones.
Postcondiciones	
Título	**Descripción**
Éxito	El sistema presenta los puntajes almacenados en el sistema.
Fracaso	La petición falla por alguna razón y no se pueden devolver los puntajes.

Tabla 3: Extensión MoB UC.3

Nombre	**UC.4:** Registrarse en el sistema
Descripción	El usuario puede registrar sus datos en el sistema y crear una cuenta.
Actor	Usuario Invitado
Flujo de Eventos	
Flujo Básico	
El caso de uso inicia cuando el usuario selecciona la opción de “registrarse”. 1. El sistema presenta un formulario. 2. El usuario rellena el formulario y acepta los términos de uso.	

3. El sistema valida y almacena los datos y envía un link de activación al correo del usuario. 4. El usuario hace clic en el link de activación. 5. El sistema valida y activa la cuenta del usuario. 6. El caso de uso termina.	
Flujo Alternativo	
Título	**Descripción**
Datos incorrectos	1. El usuario rellena el formulario y acepta los términos de uso. 2. El sistema valida los datos y encuentra un error o dato faltante. 3. El sistema envía un aviso de error al usuario.
Precondiciones	
Título	**Descripción**
N/A	N/A
Postcondiciones	
Título	**Descripción**
Éxito	El usuario se encuentra registrado en el sistema.
Fracaso	El usuario no pudo ser registrado.

Tabla 4: Extensión MoB UC.4

Nombre	**UC.5:** Identificarse en el sistema
Descripción	El usuario registrado puede identificarse en el sistema para otras funcionalidades de éste.
Actor	Usuario Registrado
Flujo de Eventos	
Flujo Básico	
El caso de uso inicia cuando el usuario activa el icono de la extensión. 1. El sistema presenta el formulario para el inicio de sesión. 2. El usuario rellena el formulario con sus datos (nombre de usuario y contraseña). 3. El sistema valida los datos del usuario. 4. El sistema cambia de pantalla a la pantalla principal del sistema.	

5. El caso de uso termina.	
Flujo Alternativo	
Título	**Descripción**
Datos erróneos	1. El usuario rellena el formulario de identificación con sus datos. 2. El sistema valida los datos y no encuentra coincidencia en el sistema. 3. El sistema arroja un error de datos incorrectos.
Cuenta no activada	1. El usuario rellena el formulario de identificación con sus datos. 2. El sistema valida los datos, encuentra coincidencia en el sistema pero no la cuenta no se encuentra activada. 3. El sistema arroja un aviso al usuario para que active su cuenta antes de continuar.
Precondiciones	
Título	**Descripción**
Usuario Registrado	El usuario debe tener una cuenta en el sistema.
Cuenta activada	La cuenta del usuario debe estar activada a través del link de activación.
Postcondiciones	
Título	**Descripción**
Éxito	El usuario se identifica exitosamente en el sistema.
Fracaso	El usuario no se pudo identificar en el sistema.

Tabla 5: Extensión MoB UC.5

Nombre	**UC.6:** Realizar segmentación manual
Descripción	El sistema le presenta una serie de herramientas al usuario identificado dándole la posibilidad de segmentar manualmente la página Web donde se encuentra.
Actor	Usuario Registrado

<table>
<tr><th colspan="2">Flujo de Eventos</th></tr>
<tr><td colspan="2">Flujo Básico</td></tr>
<tr><td colspan="2">El caso de uso inicia cuando el usuario identificado selecciona la opción de “Comenzar”.
1. El sistema le presenta un menú con diferentes acciones de segmentación manual al usuario.
2. El usuario selecciona alguna de las acciones.
3. El sistema reacciona de acuerdo a la acción escogida.
4. El caso de uso termina.</td></tr>
<tr><td colspan="2">Flujo Alternativo</td></tr>
<tr><td>Título</td><td>Descripción</td></tr>
<tr><td>Herramienta previamente iniciada</td><td>1. El usuario inicia la herramienta de segmentación manual.
2. El sistema presenta el menú de acciones.
3. El usuario vuelve a iniciar la herramienta de segmentación manual.
4. El sistema presenta un aviso de “herramienta ya ha sido iniciada”.</td></tr>
<tr><td colspan="2">Precondiciones</td></tr>
<tr><td>Título</td><td>Descripción</td></tr>
<tr><td>Usuario identificado</td><td>El usuario debe estar identificado exitosamente dentro del sistema.</td></tr>
<tr><td colspan="2">Postcondiciones</td></tr>
<tr><td>Título</td><td>Descripción</td></tr>
<tr><td>Éxito</td><td>La herramienta se inicializa exitosamente y el usuario es capaz de hacer uso de las acciones presentadas.</td></tr>
<tr><td>Fracaso</td><td>La herramienta presenta un error en su inicialización por alguna razón.</td></tr>
</table>

Tabla 6: Extensión MoB UC.6

En la figura 14 presentada a continuación se muestra un mock-up de la Interfaz para la extensión MoB.

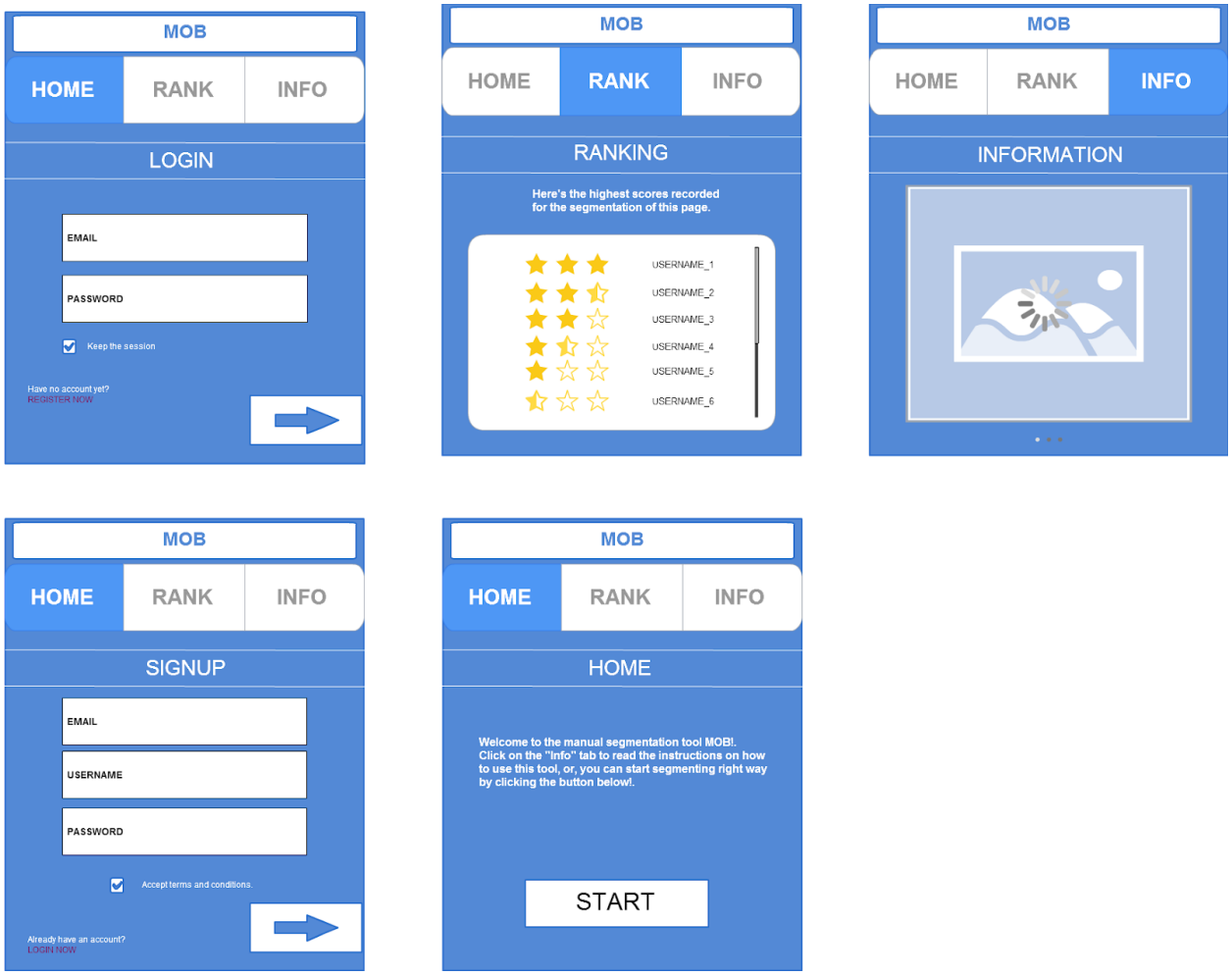

Figura 14: Mock-up Extensión de MoB

Para la interfaz se escogió un diseño minimalista basado en colores simples y la predominancia de metáforas . Se escogió una paleta de colores sencilla, basada únicamente en la combinación de dos colores principales: **#5ac2ec** (azul) y **#ffffff** (blanco) . Anexando a estos, los colores necesarios para las alertas y avisos, éxito: **#3c763d** (verde), alerta: **#ffc107** (amarillo), error: **#a94442** (rojo). En la figura 15 se muestra un ejemplo de los colores anteriormente mencionados y la interfaz del menú de la herramienta de segmentación.

Figura 15: Herramienta y Paleta de Colores

Además, se realiza un análisis para definir el modelo de granularidad de una segmentación, ya que se requiere que el usuario pueda segmentar la página Web en diferentes granularidades (de la 0 a la 10).

Se requiere que la segmentación posea un nivel de granularidad determinado, se puede ver el nivel de granularidad como el nivel de detalle de la segmentación, a menor granularidad mayor nivel de detalle.

Los factores que influyen en determinar la granularidad de la segmentación son:

La relación que existe entre el área de los bloques segmentados y el área total del documento (a mayor tamaño del bloque, mayor es el nivel de abstracción de la segmentación, por lo tanto mayor granularidad).

La cantidad de bloques que posea la segmentación (a menor número de bloques, mayor abstracción, mayor granularidad).

El nivel de granularidad puede variar inmensamente de un documento a otro, pero la relación se debe mantener: mientras mayor número de bloques, menor debe ser el tamaño de cada uno; a mayor nivel de granularidad se debe limitar el número de bloques permitido, impidiendo a su vez el decremento del área de los mismos.

En un principio se buscaba una relación meramente matemática entre las áreas, sin embargo eso ocasionaba que hubiesen grandes diferencias si el bloque cambiaba ligeramente de tamaño. Además, una relación directa haría que una segmentación con granularidad de 5, tuviese permitido un máximo de 2 bloques (cada uno con 50% del área del documento) lo cual no es lo que se desea.

Se desea:

- En cero de granularidad, se puede tener un gran número de bloques (por lo que los bloques pueden llegar a ser muy pequeños).
- En granularidad 10 se desea un solo bloque, el documento completo.
- En granularidad 9 se desea un máximo general de 3 bloques (header, content y footer).
- En el intervalo [1,8] se desea una interpolación de granularidad con los límites anteriormente descritos.

Se hace notación de que G(x) se lee como “Granularidad del documento a un nivel x”. Se tiene que a G(0) se pueden tener tantos bloques como se desee, pero en G(1) se

establece un número máximo de bloques de 40.

Se interpolan G(1) y G(9):
G(1) = 40 bloques
G(9) = 3 bloques

G(5) = (G(1) + (G9)) / 2 = 21,5
=> G(5) = 22

Se interpolan G1(1) con G(5) y G(5) con G(9):
G(1) = 40 bloques
G(5) = 22 bloques
G(9) = 3 bloques

G(3) = (G(1) + (G5)) / 2 = 31
=> G(3) = 31

G(7) = (G(5) + (G9)) / 2 = 12,5
=> G(7) = 13

Se interpolan G(1) con G(3), G(3) con G(5), G(5) con G(7) y G(7) con G(9):
G(1) = 40 bloques
G(3) = 31 bloques
G(5) = 22 bloques
G(7) = 13 bloques
G(9) = 3 bloques

G(2) = (G(1) + (G3)) / 2 = 35,5
=> G(2) = 36

G(4) = (G(3) + (G5)) / 2 = 26,5
=> G(4) = 27

G(6) = (G(5) + (G7)) / 2 = 17,5
=> G(6) = 18

G(8) = (G(7) + (G9)) / 2 = 8
=> G(8) = 8

Con esto se obtiene el máximo de bloques para cada nivel de granularidad, Como esto representa el número máximo de “pedazos” en los que se puede dividir el documento, lo lógico sería considerar que el área máxima que cada uno de estos bloques puede llegar a alcanzar representa el área mínima que debe poseer los bloques en este nivel de granularidad, ya que pueden existir menos bloques de los esperados, pero no más.

Se puede calcular el área mínima que debe poseer cada bloque en cada una de las granularidades para una página determinada, esto permite poder clasificar los diferentes bloques dentro de una determinada granularidad, por ejemplo, si el bloque posee un área que se encuentra entre el área mínima de la granularidad 4 y la granularidad 5, se dice que el bloque es de granularidad 4.

Lo ideal sería que la segmentación posea únicamente bloques que posean la misma granularidad que la segmentación, sin embargo, ese no siempre resulta el caso y dado que esto puede resultar muy restrictivo, se decide dar un rango de error de una (1) granularidad, es decir, el bloque aún es aceptado si se encuentra una granularidad por debajo de la granularidad de la segmentación, es decir, si el área del bloque es mayor o igual que el área del documento dividido entre el factor de granularidad de la segmentación menos 1.

$$A_{bi} \geq \frac{A_{doc}}{G(x-1)}$$

- **Colaboración:**

En la etapa de colaboración de este ciclo se desarrollaron los requerimientos descritos en la etapa anterior haciendo uso del lenguaje de marcado **HTML5** y las reglas de estilos **CSS3** (*c.f* Capítulo 2.1.8.1) para la presentación de la interfaz de la extensión y la herramienta de segmentación, además del lenguaje de programación **Javascript** (*c.f* Capítulo 2.1.8.2) para realizar la conección con la API, y manejar los diferentes eventos que se describieron en los casos de usos para la recolección de datos y control de la sesión.

La lógica que se oculta tras la extensión es la siguiente:

1. El usuario se registra e inicia sesión en la extensión. Al iniciar sesión le aparece la opción para iniciar la herramienta de segmentación manual.

2. El usuario inicia la herramienta de segmentación manual, en el fondo, el Javascript del la extensión inyecta el Javascript de la herramienta en la página donde se encuentra el usuario en ese momento y envía un mensaje mediante el API de Chrome[10].
3. El mensaje es captado por la ventana activa (aquella en la que se encuentra el usuario) y se inicializa la herramienta de segmentación, comenzando por la importación de las librerías auxiliares, la captura de pantalla de la ventana (usando el API de Chrome) y activando los listeners de Javascript necesarios para el control de las acciones del menú de la herramienta.

Entre las funcionalidades del menú de la herramienta de segmentación se encuentran:

- **Agregar nuevo bloque:**

Permite agregar un nuevo bloque a la segmentación, al ser seleccionada el usuario puede recorrer los elementos del DOM con el mouse y estos serán iluminados, una vez el usuario haga click sobre alguno de estos elementos del DOM, se inyectará en el HTML un cuadrado representando el bloque segmentado, dicho elemento posee entre sus atributos los datos del bloque segmentado (etiqueta, ancho, alto, posición en left, posición en top, área), dichos datos son también almacenados dentro de un arreglo en Javascript. Este arreglo es el que es utilizado para realizar las comparaciones necesarias entre los diferentes bloques para llevar a cabo las futuras acciones.

Cabe destacar que al realizar esta acción se activa un subproceso que verifica si hay otros bloques dentro del que está a punto de crearse, en cuyo caso se eliminan dichos bloques y se conserva únicamente el nuevo que ha sido creado. En caso de que existan bloques que intercepten mas no se encuentren completamente dentro del nuevo, ocurriría un error de intercepción el cual requerirá de una acción de “cortar”. En la figura 16 se puede ver un ejemplo de esta acción en uso, el bloque resultante está en un estado de error por defecto pues se debe etiquetar.

[10] Google presenta un API para la comunicación entre el navegador, las extensiones y las páginas Web. Puede consultar la documentación aquí: https://developer.chrome.com/extensions/devguide

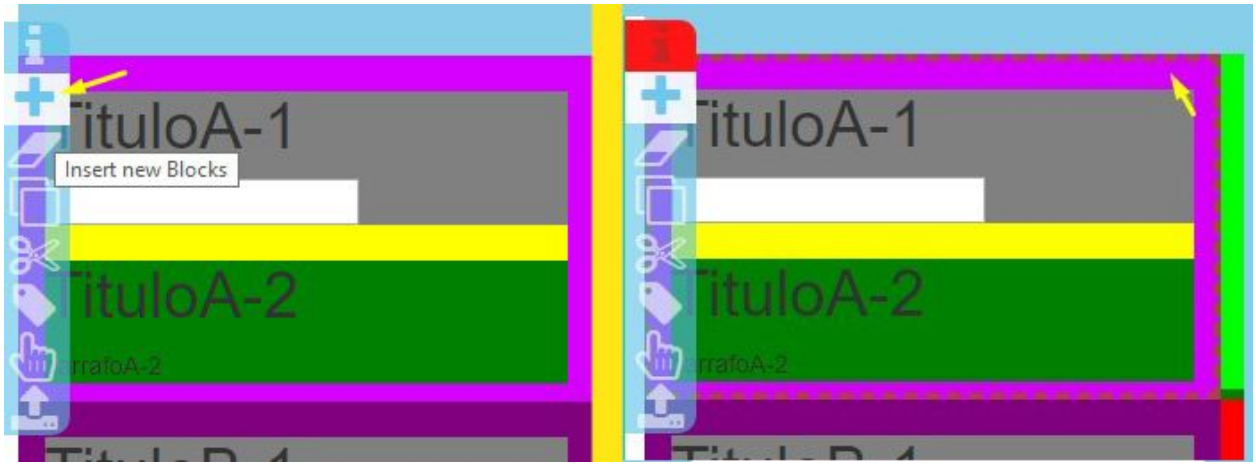

Figura 16: Acción de Crear Bloque

- **Eliminar bloque:**

Al estar esta acción activada, se iluminaran aquellos bloques al que el usuario señale con el mouse, al hacer click sobre alguno de estos bloques se disparará una función de Javascript la cual removerá el elemento DOM que representa el rectángulo del bloque, y dicho bloque será removido del arreglo de bloques, observar la figura 17.

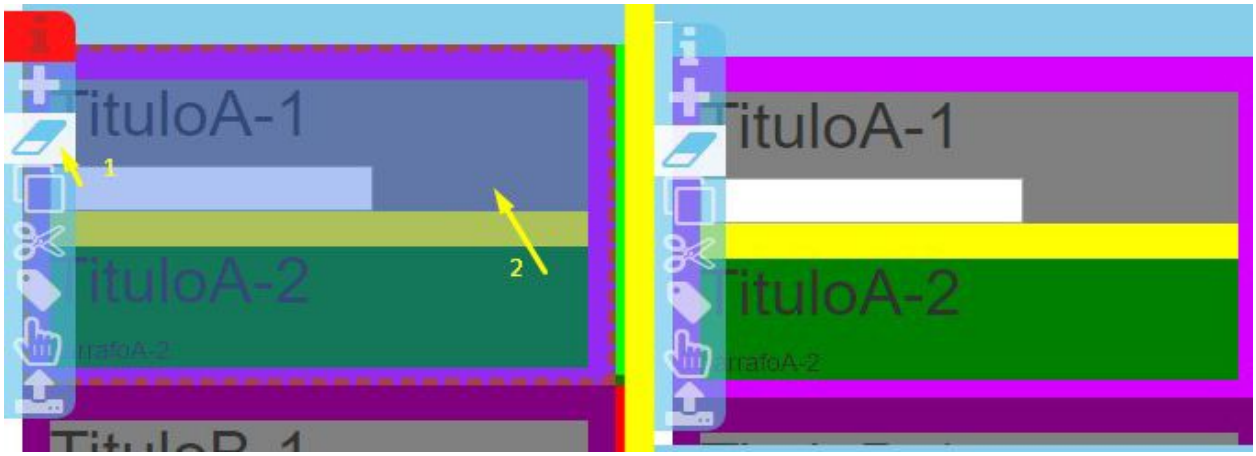

Figura 17: Acción Eliminar Bloque

- **Unir bloques:**

Al ser seleccionada, esta acción permite al usuario seleccionar dos bloques los cuales se unirán. Esta acción elimina los dos bloques seleccionados y crea uno nuevo que comparte los límites superiores e inferiores máximos de los bloques anteriores. Después se comprueba que no hayan quedado atrapados otros bloques dentro del nuevo, en cuyo caso se eliminarían, como se muestra en la figura 18.

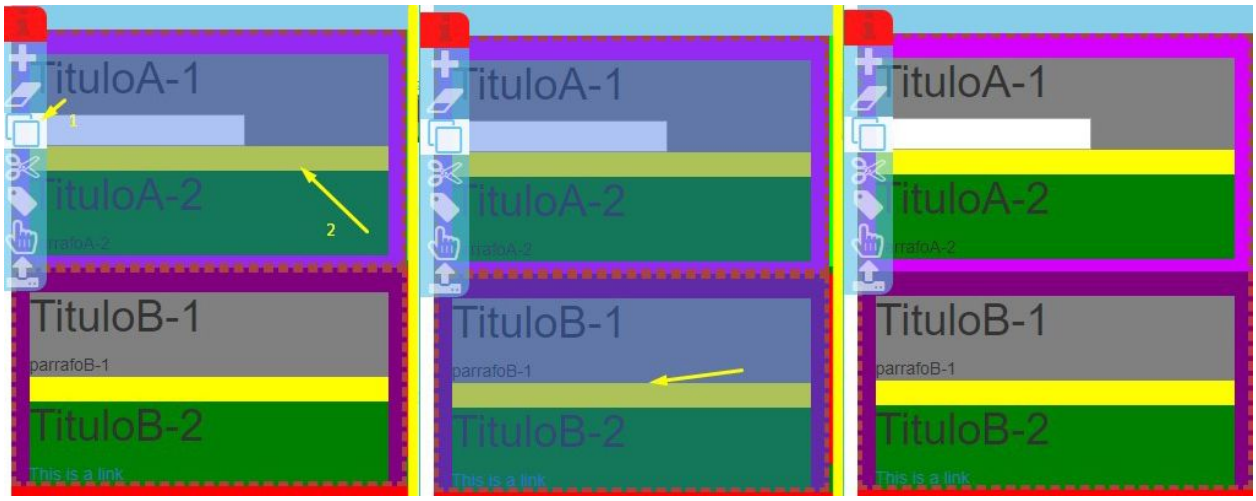

Figura 18: Acción Unir Bloques

- **Cortar bloques:**

Esta acción permite seleccionar dos bloques que se están interceptando (A y B) y realizar un corte entre los dos. El orden de selección importa, pues A será el bloque que predominará (se mantendrá intacto) y B será el bloque que se recortará para solucionar la intercepción. Un ejemplo de este proceso se muestra en la figura 19.

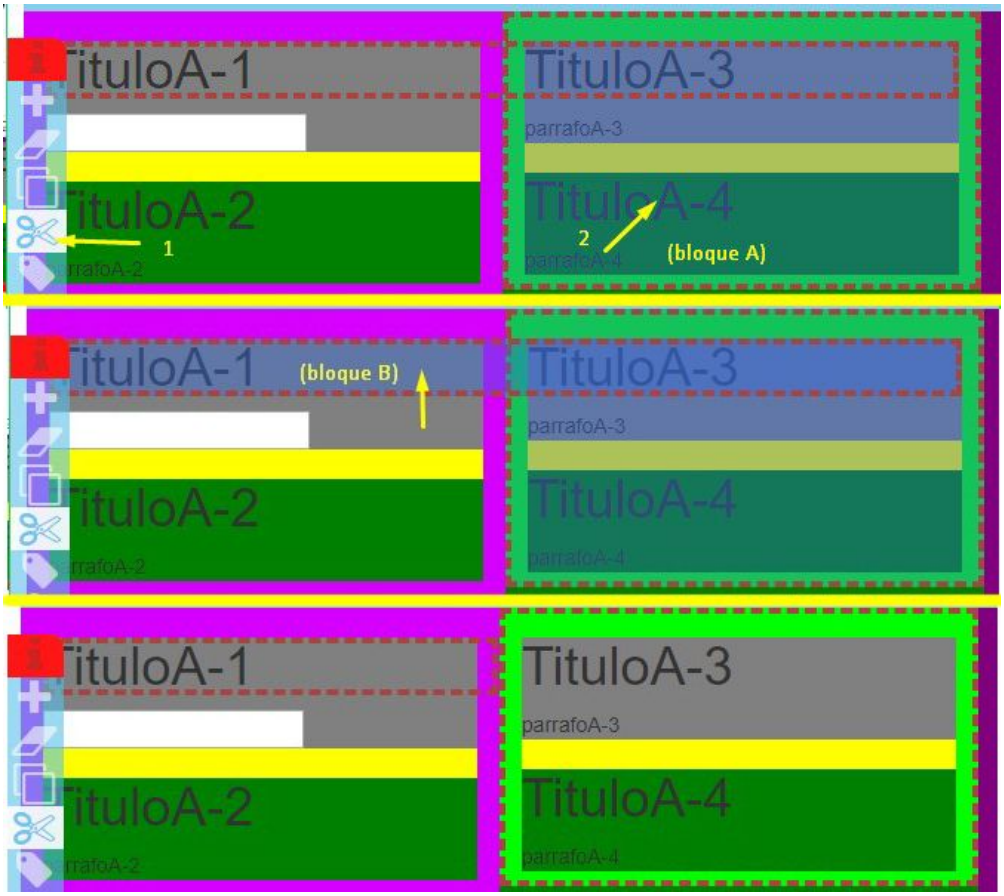

Figura 19: Acción Cortar Bloques

- **Etiquetar bloque:**

Al ser seleccionada, esta acción le permite al usuario seleccionar cualquier bloque presente en la segmentación, al hacer click sobre un bloque se mostrará una modal con una lista donde el usuario podrá escoger la etiqueta que mejor se adapte al bloque. En la figura 20 se muestra el ejemplo de esta acción.

Figura 20: Acción Etiquetar Bloque

- **Seleccionar bloque:**

Al estar seleccionada, esta acción permite al usuario seleccionar cualquier bloque y obtener una ventana de información con los datos de dicho bloque. En la figura 21 se muestra un ejemplo de esta acción.

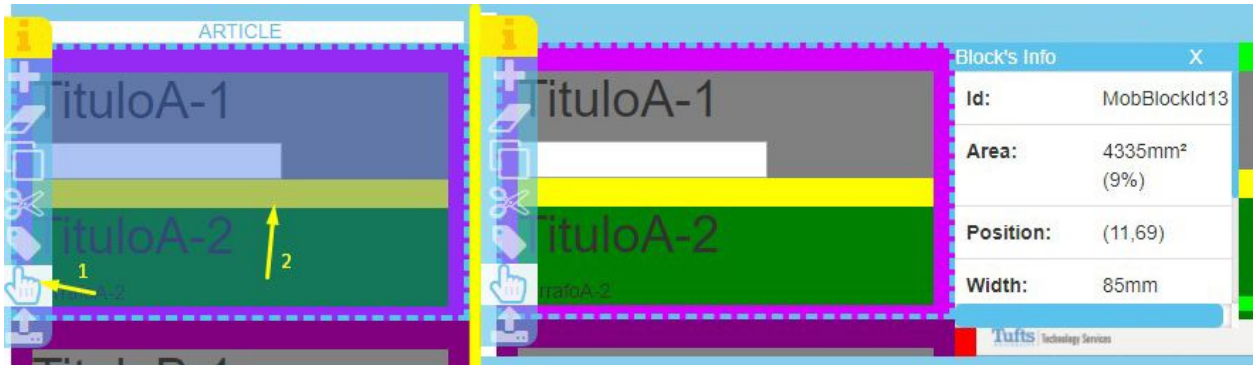

Figura 21: Acción Seleccionar Bloque

- **Panel de información:**

Al ser activada, esta acción despliega un panel informativo con metadatos de la segmentación para el usuario, ofrece la opción de cambiar la granularidad de la segmentación y muestra todas las alertas que puede presentar la segmentación. En la figura 22 se presenta una captura de dicho panel.

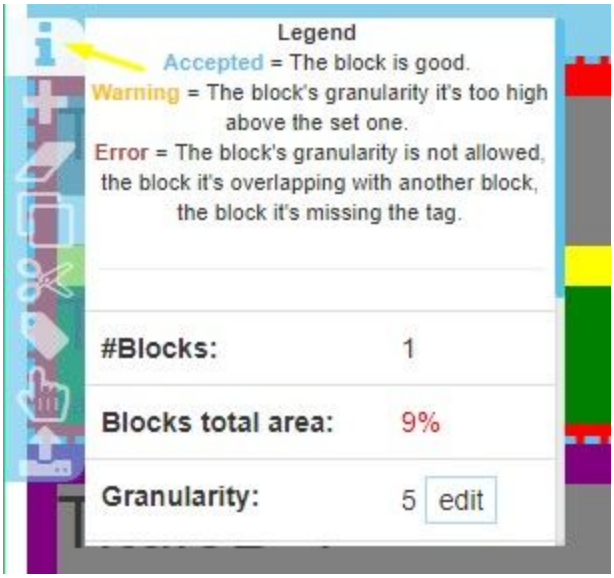

Figura 22: Acción Panel de Información

- **Enviar segmentación:**

Esta acción activa los procesos necesarios para la recolección de datos y envío hacia el API de MoB, comienza por comprobar el estado de la segmentación en busca de errores o advertencia, en caso de presentar errores, la segmentación no será enviada y los procesos de recolección de datos se cancelan, en caso contrario, se crea una estructura de json para enviar todos los datos necesarios al API, para poder obtener todos los datos, se almacena el HTML actual de la página en un string, esto vendría siendo el HTML de la página versión MoB. A la vez se hace una búsqueda entre los elementos DOM de la página para identificar cuales son los elementos DOM que están presente en cada bloque segmentado, esta información se almacena en un arreglo junto con los bloques de la segmentación.

- **Aprendizaje:**

Durante la realización de este ciclo se llevó a cabo la decisión de dejar el framework de AngularJS de lado y realizar la funcionalidad SPA de la interfaz meramente en Javascript (también conocido como Vanilla Javascript, pues no se requirió de ningún framework), esto motivado a la sencillez que se quería mantener en la interfaz.

A pesar de que no formaba parte de los requerimientos originales, se decidió desarrollar la toma de una captura de la página Web para poder presentarla en el Repositorio de MoB de una forma más agradable para el usuario. De igual forma se anexó el panel de información y la acción de “seleccionar bloque” para ofrecer información sobre la segmentación al usuario que la estuviese realizando.

Objetivos logrados:

- Desarrollar funcionalidades de agregar, editar, eliminar y etiquetar bloques de segmentos para permitir al usuario segmentar visualmente una página web.
- Desarrollar una funcionalidad que permita tomar la segmentación visual realizada por el usuario y transformarla en un árbol de segmentación basado en el árbol DOM de la página web segmentada.
- Desarrollar una funcionalidad que permita tomar la segmentación visual realizada por el usuario y obtener el árbol estructural de bloques (rectangulares) asociados al diseño de la página.

- Implementar funcionalidades de registro e identificación de usuarios, las cuales permitan el control y manejo de sesión de usuarios.

A continuación se muestra una breve comparación entre la antigua herramienta MoB y la obtenida con el desarrollo de este trabajo.

En la figura 23 se puede observar que la nueva herramienta posee las siguientes características a diferencia de la antigua versión:

- El color de los bloques refleja el estado de los mismos y no el nivel (pues la segmentación es plana).
- El panel de información ofrece mayor información sobre la granularidad presente y los bloques además de los posibles errores o advertencias que puedan ocurrir.
- La caja de herramienta ocupa mucho menos espacio al estar conformada únicamente de metáforas de las acciones, también es semi-transparente para poder observar el contenido que hay detrás.
- Todas las acciones son llevadas a cabo únicamente con el mouse, sin tener que usar el teclado.
- Se ofrece la acción de “cortar” para poder separar bloques que se intersectan, y la acción “seleccionar” que permite obtener toda la información de un bloque en específico.

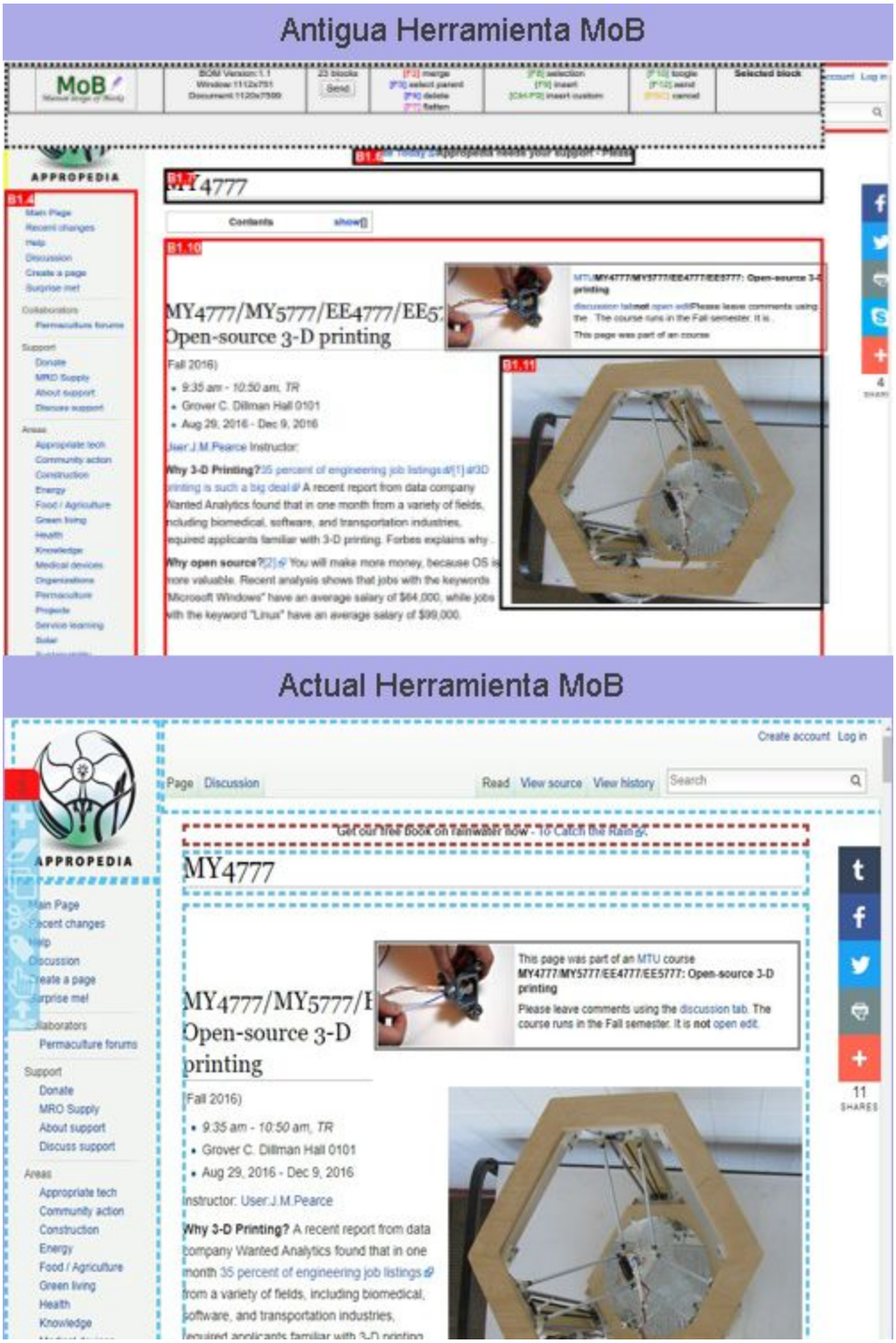

Figura 23: Comparación de herramientas MoB

4.3.2. 2do Ciclo: MoB API

Este ciclo consiste en el desarrollo de la API de MoB, la cual además de ofrecer servicios como API también funciona como backend para el sitio Web del Repositorio de MoB y realiza la conección con la base de datos para poder almacenar y recuperar los datos de las segmentaciones.

- **Especulación:**

Durante el proceso de especulación de este ciclo se pudieron establecer los diferentes servicios que serían necesarios para satisfacer las de la extensión MoB, y otros posibles servicios que podrían llegar a ser de utilidad en el futuro. Para ello se realizó un diagrama con los modelos que conforman la base de datos. En la figura 24 se muestra un modelo Entidad-Relación para demostrar las diferentes tablas usadas y sus relaciones.

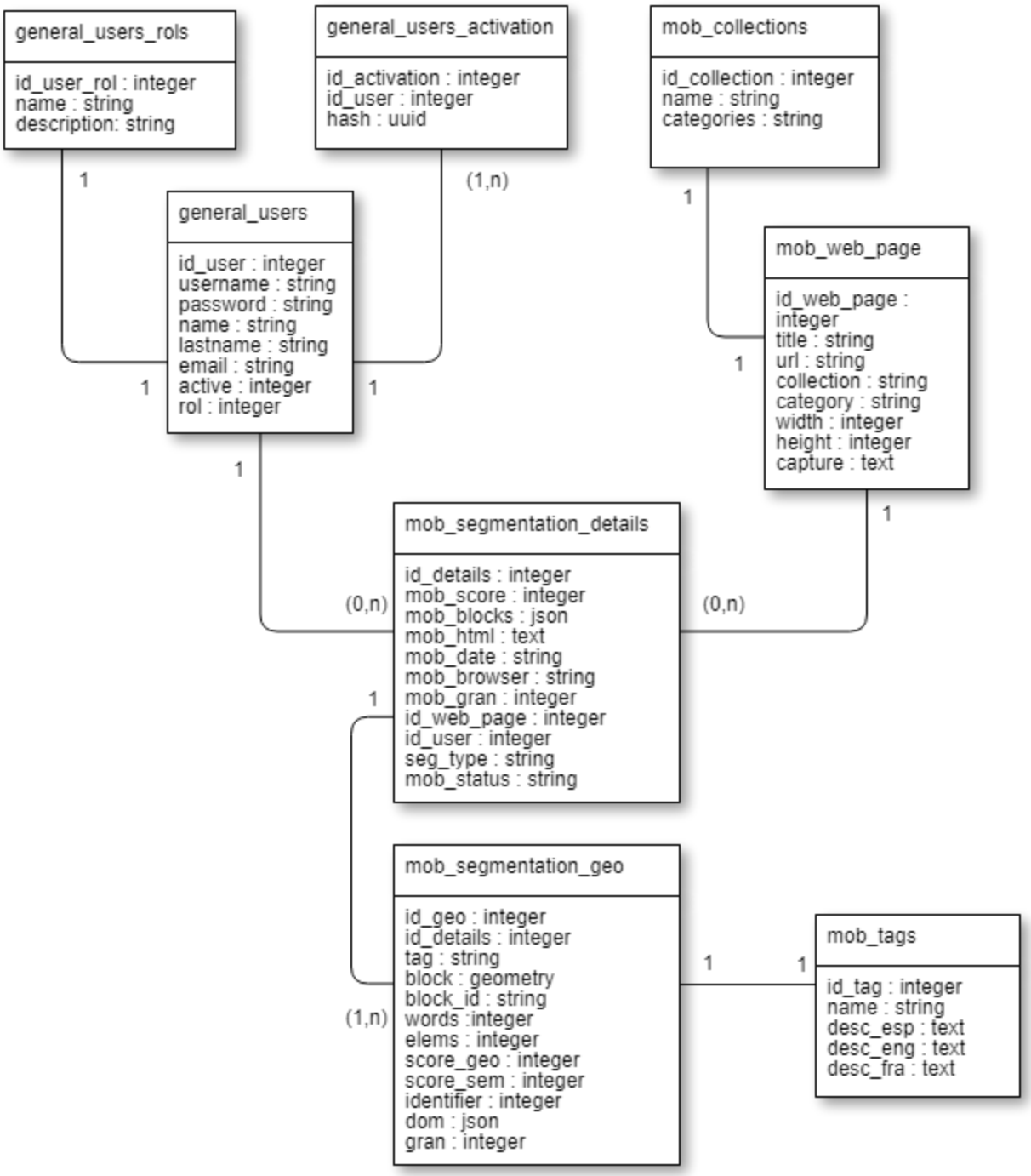

Figura 24: Modelo Entidad-Relación de las Tablas en el Sistema

Una breve explicación del modelo mostrado anteriormente: en general, en el sistema existen 3 grandes entidades, el usuario, la página Web y la segmentación. La tabla del usuario (general_users) se apoya en dos tablas, una para los roles del usuario (general_users_rols) y otra para la activación del mismo (general_users_activation). La

tabla de la página Web (mob_web_page) se apoya en una tabla para indicar su colección y categoría (mob_collections). Finalmente, la tabla de la segmentación (mob_segmentation_details) se apoya de una tabla que representa los bloques de la segmentación (mob_segmentation_geo), dichos datos se encuentran de igual forma en el atributo “mob_blocks” de la tabla de la segmentación, sin embargo, se crea la tabla de mob_segmentation_geo para poder realizar los análisis con el complemento de Postgis.

- **Colaboración:**

El desarrollo de la API se realizó con el lenguaje de programación **Python v.3.5**, apoyado con el microframework **Flask v.0.12.2** (*c.f* Capítulo 2.1.8.3). Para la creación de la base de datos que va conectada a la API se utilizó el manejador de base de datos **Postgresql v.10.1** junto con un componente llamado **Postgis 2.4** (*c.f* Capítulo 2.1.8.4) para realizar las operaciones entre tablas. Para conocer el proceso de instalación del ambiente, así como los componentes necesarios para su correcto funcionamiento consulte Anexos 1.

En general, el API de MoB se divide en dos partes: todos los servicios RESTful que pueden ser ofrecidos a la herramienta de MoB o a terceros y todas aquellas funciones que manejan el backend del Repositorio MoB.

A continuación se explican los servicios RESTful que son ofrecidos por la API, mientras que las funciones que manejan el backend del Repositorio de MoB se explicarán en el siguiente ciclo. Los servicios ofrecidos son los siguientes:

- **Registrar usuario:** Permite registrar a un usuario en el sistema, para completar el registro se le envía al usuario un link de activación a su correo. Esta funcionalidad de envío de correos se logra gracias al módulo de “flask_mail”.

- **Iniciar sesión:** Permite al usuario registrado (y activado) iniciar sesión en el sistema para hacer uso de sus funcionalidades. Se mantiene la sesión del lado del servidor usando la variable “Session” de Flask.

- **Cerrar sesión:** Borra las cookies de sesión existentes en el navegador y la sesión existente en el API.

- **Recuperar contraseña:** Permite al usuario recuperar su contraseña en caso de extravío, el sistema envía una combinación aleatoria de caracteres como contraseña temporal dado a que por medidas de seguridad las contraseñas se encuentran encriptadas por hash en la base de datos.

- **Obtener colecciones:** Permite obtener una lista con los nombres de las colecciones y categorías de éstas existentes en la base de datos del sistema.

- **Obtener etiquetas:** Permite obtener una lista con los nombres de las etiquetas existentes en la base de datos del sistema.

- **Obtener puntajes globales:** Permite obtener una lista con los mejores puntajes en cada una de las granularidades de una página Web específica.

- **Obtener puntajes del usuario:** Permite obtener una lista con los puntajes en cada una de las granularidades de una página Web específica para un usuario determinado.

- **Cargar segmentación:** Este es uno de los servicios más importante del API pues representa la base de todo el sistema, permite cargar los resultados de una segmentación a la base de datos (y los datos de la página Web en caso de que sea la primera vez que se segmenta). Formando de esta forma lo que denominamos anteriormente como *ground truth*.

- **Vista previa de segmentación:** Devuelve un canvas con las figuras y etiquetas de los bloques segmentados para una segmentación en específico, también se puede especificar el nivel de “zoom” o tamaño del canvas.

- **Obtener segmentación en formato JSON:** Retorna un JSON con todos los datos de una segmentación en específico.

- **Obtener segmentación en formato V-PRIMA:** Retorna todos los datos de una segmentación específica en formato V-PRIMA, el formato V-PRIMA consta de un XML donde se especifican los bloques existentes en la segmentación y los links, imágenes y textos existentes dentro de éstos..

- **Obtener segmentación en formato MoB HTML:** Dada una segmentación determinada, retorna un HTML con la información que se capturó momentos

antes de enviar la segmentación, es decir, el HTML original de la página Web modificado por la herramienta MoB tras realizar la segmentación.

- **Obtener página Web en formato WARC:** Devuelve la información de una página Web en formato WARC (Web ARChive), el formato WARC permite la concatenación de múltiples objetos de datos o recursos en un solo archivo, de esta forma es utilizado para almacenar la información de páginas Web junto con sus recursos y metadata. Para lograr esto se utilizó la utilidad "wget" que viene incluida en el sistema de Ubuntu/Linux de la siguiente forma:

```
wget --no-check-certificate --warc-file='ruta_donde_se_guardara' --recursive --level=1
-O tempfile 'url_de_la_pagina'
```

Dado que es un comando que debe usarse en una consola de comandos y no desde Python, fue necesario usar el módulo de "subprocess' de Python para poder correr el comando.

Para más información acerca de los servicios ofrecidos por el API de MoB y cómo utilizarlos, por favor revisar documentación anexa (*c.f* Anexos 2).

- **Aprendizaje:**

Durante el desarrollo de este ciclo se decidió dejar a un lado la base de datos MongoDB y usar en cambio Postgresql dado a que esta posee el componente Postgis, el cual facilita las operaciones necesarias para obtener la mejor segmentación, y en un futuro facilitará las operaciones necesarias para la comparación entre segmentaciones. Esto gracias a las funciones integradas de Postgis que facilitan las operaciones geométricas (en el caso de este trabajo, los rectángulos) directamente desde las tablas de la base de datos, ahorrando así el trabajo de crear estas funciones desde cero.

Además se suplanta el framework de Django por Flask, por el motivo de que al ser un API relativamente sencilla y al manejar un backend (el del Repositorio) relativamente pequeño se consideró que un framework tan completo y complejo como Django no sería necesario, sino que bastaría con algo mucho más sencillo y rápido de programar como el microframework de Flask.

Cabe acotar que también se anexó el desarrollo de servicios extras para satisfacer posibles futuras necesidades del sistema, como lo fueron: la vista previa de

la segmentación, obtener la segmentación en diferentes formatos (JSON, MoB HTML, V-PRIMA) y obtener la página Web en formato WARC.

Objetivos logrados:

- Implementar una Base de Datos que almacene todos los resultados obtenidos por la herramienta y los datos de los usuarios que hacen uso de ella.
- Implementar funcionalidades en el lado del servidor tal que permitan el análisis y la visualización de la data almacenada.
- Conformar un *ground truth* a partir de los datos obtenidos por la herramienta de segmentación manual.

4.3.3. 3er Ciclo: MoB Repository (Repositorio MoB)

En este ciclo se desarrolló el sitio Web llamado MoB Repository, el cual tiene como finalidad ofrecer una interfaz usable para que los usuarios puedan observar todas las segmentaciones almacenadas en el sistema, también ofrece control a administradores para editar la lista de etiquetas, colecciones y categorías, entre otras funciones que se describirán a continuación.

- **Especulación:**

Durante la etapa de especulación de este ciclo se realizó un diagrama de casos de usos para determinar cuáles iban a ser los requerimientos del Repositorio. En las figuras 25, 26, 27 y 28 se muestran los diagramas de caso de usos resultantes, seguido por la tabla 7 hasta la tabla 25, correspondientes a la descripción de los casos de usos.

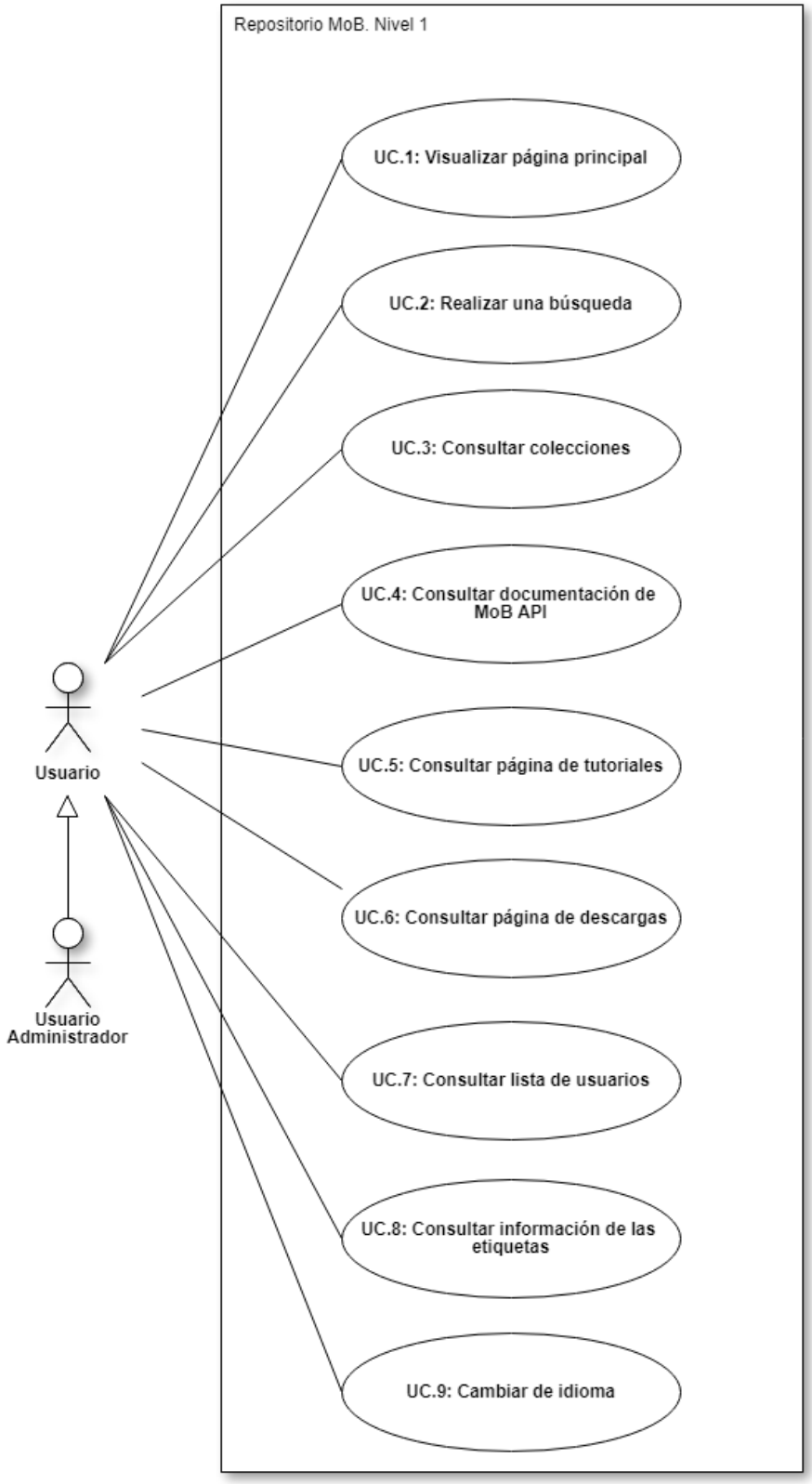

Figura 25: Diagrama de Casos de Uso para el componente Mob Repository, Nivel 1

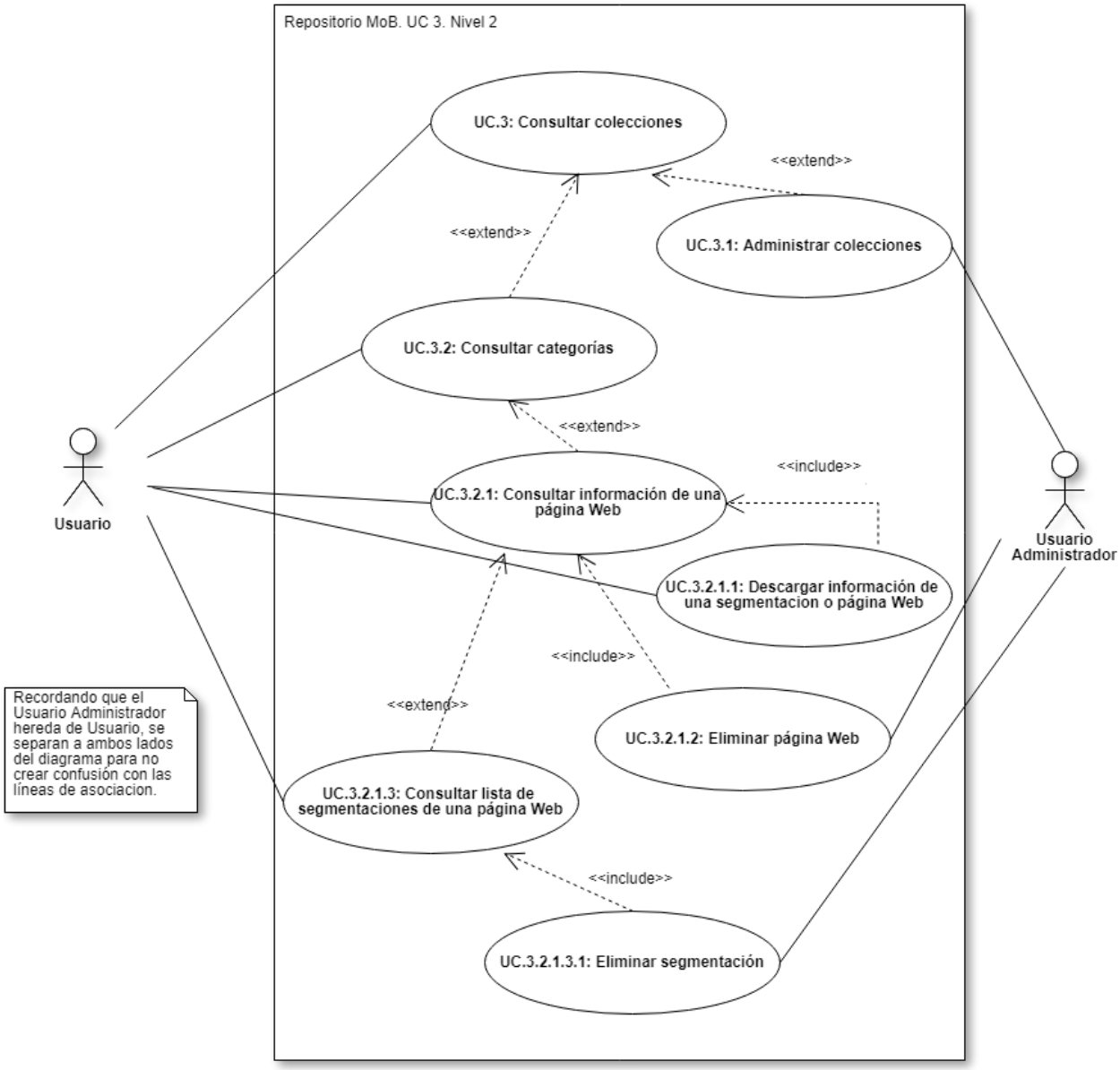

Figura 26: Diagrama de Casos de Uso 3 para el componente Mob Repository, Nivel 2

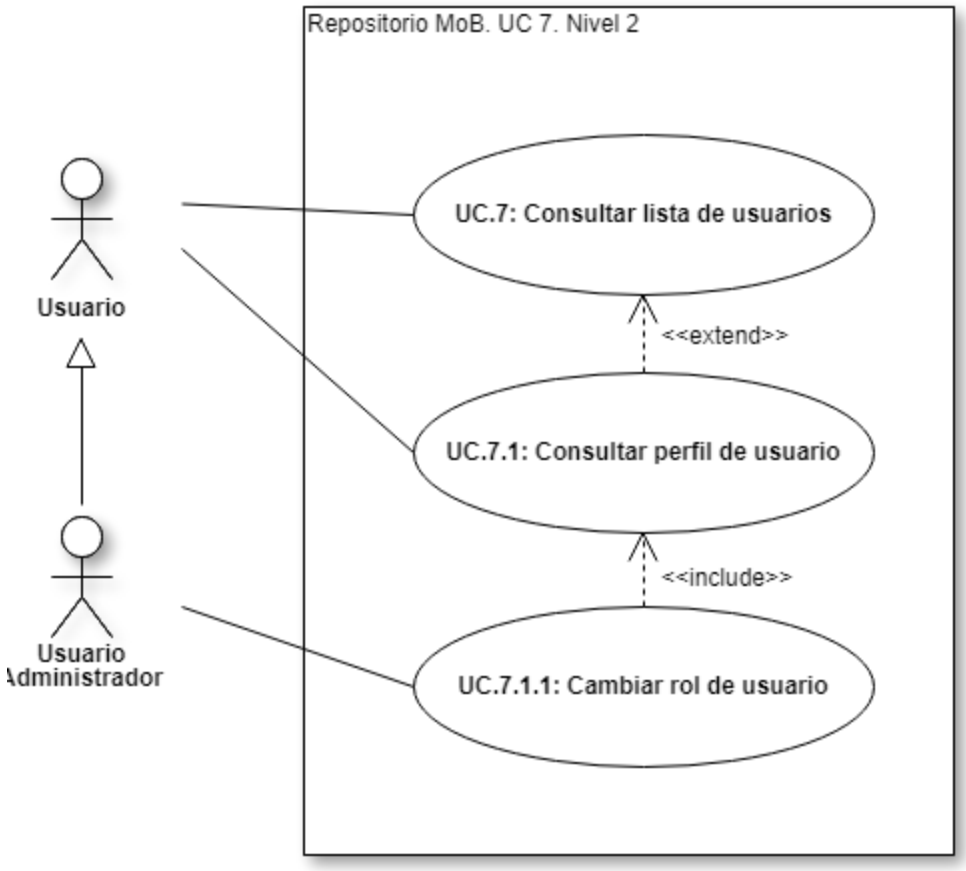

Figura 27: Diagrama de Casos de Uso 7 para el componente Mob Repository, Nivel 2

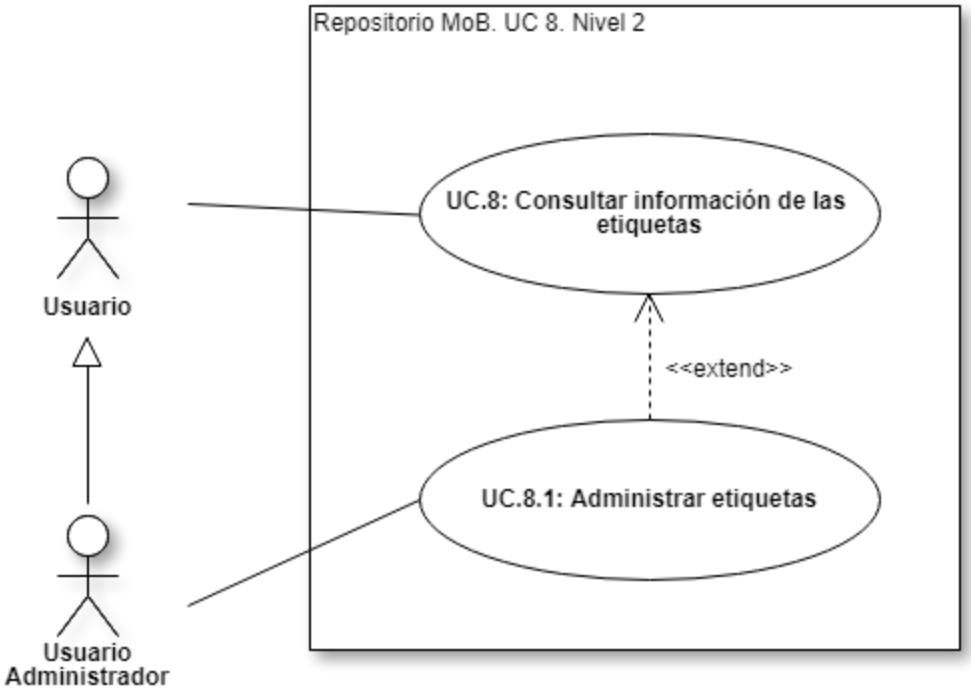

Figura 28: Diagrama de Casos de Uso 8 para el componente Mob Repository, Nivel 2

Nombre	**UC.1:** Visualizar página principal
Descripción	El usuario puede observar/consultar la información de bienvenida que aparece en la página principal.
Actor	Usuario y Usuario Administrador
Flujo de Eventos	
Flujo Básico	
El caso de uso inicia apenas el usuario entra en el sistema. 1. El sistema le presenta al usuario información relevante y concreta sobre el sistema y sus componentes. 2. El caso de uso termina.	
Flujo Alternativo	
Título	**Descripción**
N/A	N/A
Precondiciones	
Título	**Descripción**
N/A	N/A
Postcondiciones	
Título	**Descripción**
Éxito	El sistema devuelve información de la página de bienvenida.
Fracaso	El sistema devuelve un mensaje de error.

Tabla 7: Repositorio MoB UC.1

Nombre	**UC.2:** Realizar una búsqueda
Descripción	El usuario puede realizar una búsqueda en el sistema.
Actor	Usuario y Usuario Administrador
Flujo de Eventos	
Flujo Básico	
El caso de uso inicia apenas el usuario entra en el sistema. 1. El usuario rellena el formulario de la barra de búsqueda.	

2. El sistema valida los datos y busca las páginas Web que se asocian a éstos. 3. El sistema muestra las diferentes páginas Web que forman parte del resultado de búsqueda. 4. El caso de uso termina.	
Flujo Alternativo	
Título	**Descripción**
Todas las páginas	1. El usuario simplemente le da clic al botón para buscar sin rellenar ni especificar nada. 2. El sistema muestra todas las páginas Web almacenadas.
Precondiciones	
Título	**Descripción**
N/A	N/A
Postcondiciones	
Título	**Descripción**
Éxito	El sistema devuelve páginas Web
Fracaso	El sistema devuelve un mensaje de error.

Tabla 8: Repositorio MoB UC.2

Nombre	**UC.3:** Consultar colecciones
Descripción	El usuario puede observar las diferentes colecciones existentes en el sistema.
Actor	Usuario y Usuario Administrador
Flujo de Eventos	
Flujo Básico	
El caso de uso inicia apenas el usuario da clic en “colecciones”. 1. El sistema presenta la lista de colecciones existentes en el sistema. 2. El caso de uso termina.	
Flujo Alternativo	
Título	**Descripción**
N/A	N/A

Precondiciones	
Título	**Descripción**
Existencia	Debe haberse creado al menos una colección en el sistema.
Postcondiciones	
Título	**Descripción**
Éxito	El sistema devuelve una lista de colecciones.
Fracaso	El sistema devuelve un mensaje de error.

Tabla 9: Repositorio MoB UC.3

Nombre	**UC.3.1:** Administrar colecciones
Descripción	El usuario identificado con rol de administrador puede realizar diversas tareas de administración (editar, eliminar, crear) sobre las colecciones.
Actor	Usuario Administrador
Flujo de Eventos	
Flujo Básico	
El caso de uso inicia cuando el usuario selecciona “Editar colecciones” en la página de colecciones. 1. El sistema presenta una tabla con todas las colecciones existentes y las opciones de administrador. 2. El usuario selecciona una de las acciones de administrador. 3. El sistema aplica los cambios. 4. El caso de uso termina.	
Flujo Alternativo	
Título	**Descripción**
Crear colección	1. El usuario escoge crear una colección 2. El sistema le presenta un formulario. 3. El usuario rellena el formulario y envía. 4. El sistema valida los datos y agrega la colección a la lista.

Eliminar colección	1. El usuario escoge eliminar una colección. 2. El sistema le presenta un aviso de verificación 3. El usuario acepta 4. El sistema elimina la colección de la lista.
Editar colección	1. El usuario escoge editar una colección. 2. El sistema le presenta un formulario completado con los datos preexistentes de la colección. 3. El usuario edita los datos y envía. 4. El sistema valida los datos y actualiza la información en la lista.
Precondiciones	
Título	**Descripción**
Usuario identificado	El usuario debe estar identificado en el sistema.
Usuario administrador	El usuario debe poseer el rol de administrador.
Postcondiciones	
Título	**Descripción**
Éxito	La información de la lista de colecciones se actualiza exitosamente de acuerdo al cambio realizado por el usuario.
Fracaso	La acción de administración no se realiza por alguna razón.

Tabla 10: Repositorio MoB UC.3.1

Nombre	**UC.3.2:** Consultar categorías
Descripción	El usuario puede consultar las diferentes categorías presentes en una colección.
Actor	Usuario y Usuario Administrador
Flujo de Eventos	
Flujo Básico	
El caso de uso inicia apenas el usuario escoge una colección de la lista de colecciones. 1. El sistema presenta una lista de categorías existentes dentro de la colección. 2. El caso de uso termina.	

Flujo Alternativo	
Título	**Descripción**
N/A	N/A
Precondiciones	
Título	**Descripción**
Existencia	Debe haberse creado al menos una categoría dentro de la colección.
Postcondiciones	
Título	**Descripción**
Éxito	El sistema devuelve la lista de categorías asociadas a la colección escogida.
Fracaso	El sistema devuelve un mensaje de error.

Tabla 11: Repositorio MoB UC.3.2

Nombre	**UC.3.2.1:** Consultar información de una página Web
Descripción	El usuario puede observar la información de una página Web, así como los resultados de las segmentaciones de esa página en sus diferentes granularidades y por diferentes métodos.
Actor	Usuario y Usuario Administrador
Flujo de Eventos	
Flujo Básico	
El caso de uso inicia cuando el usuario selecciona alguna de las páginas Web de la lista que le ofrece el sistema. 1. El sistema muestra información relevante de la página Web, y una lista de las granularidades disponibles. 2. El usuario escoge una de las granularidades. 3. El sistema muestra los diferentes métodos/algoritmos realizados sobre esa granularidad. 4. El usuario escoge un método/algoritmo. 5. El sistema muestra la información de la segmentación. 6. El caso de uso termina.	
Flujo Alternativo	

Título	Descripción
N/A	N/A
Precondiciones	
Título	**Descripción**
N/A	N/A
Postcondiciones	
Título	**Descripción**
Éxito	El sistema arroja información de la página Web y sus segmentaciones.
Fracaso	El sistema arroja un mensaje de error.

Tabla 12: Repositorio MoB UC.3.2.1

Nombre	**UC.3.2.1.1:** Descargar información de una segmentación o página Web.
Descripción	El usuario puede descargar la información de una segmentación o una página Web en sus diferentes formatos disponibles.
Actor	Usuario y Usuario Administrador
Flujo de Eventos	
Flujo Básico	
El caso de uso inicia cuando el usuario entra en la página de información de una página Web. 1. El sistema le presenta diferentes botones con opciones para descargar la información de la página Web o de las segmentaciones asociadas a ésta. 2. El usuario selecciona el botón con el formato deseado. 3. El sistema responde con la descarga de un archivo que contiene el formato seleccionado por el usuario. 4. El caso de uso termina.	
Flujo Alternativo	
Título	**Descripción**
N/A	N/A
Precondiciones	

Título	**Descripción**
N/A	N/A
Postcondiciones	
Título	**Descripción**
Éxito	El usuario posee el archivo con la información y formato deseado.
Fracaso	El archivo no pudo ser descargado por alguna razón.

Tabla 13: Repositorio MoB UC.3.2.1.1

Nombre	**UC.3.2.1.2:** Eliminar página Web
Descripción	El usuario identificado como administrador puede eliminar una página Web y todas las segmentaciones de la misma del sistema.
Actor	Usuario Administrador
Flujo de Eventos	
Flujo Básico	
El caso de uso inicia cuando el usuario selecciona “eliminar página Web” desde la página informativa de la página Web. 1. El sistema levanta una alerta de confirmación. 2. El usuario confirma. 3. El sistema elimina todo la información de la página Web y de todas las segmentaciones asociadas a ésta del sistema. 4. El caso de uso termina.	
Flujo Alternativo	
Título	**Descripción**
Cancelar	1. El sistema levanta una alerta de confirmación. 2. El usuario cancela la acción.
Precondiciones	
Título	**Descripción**
Usuario identificado	El usuario debe estar identificado en el sistema.
Usuario administrador	El usuario debe poseer el rol de administrador.
Postcondiciones	

Título	Descripción
Éxito	El sistema devuelve al usuario a la lista de páginas Web con un mensaje de éxito.
Fracaso	El sistema devuelve un mensaje de error.

Tabla 14: Repositorio MoB UC.3.2.1.2

Nombre	**UC.3.2.1.3:** Consultar lista de segmentaciones de una página Web.
Descripción	El usuario puede visualizar la lista de segmentaciones que conforman el *ground truth* de una página Web.
Actor	Usuario y Usuario Administrador
Flujo de Eventos	
Flujo Básico	
El caso de uso inicia apenas el usuario da clic en "*ground truth*" en la página de información de la página Web. 1. El sistema presenta una lista de las segmentaciones asociadas a la página Web. 2. El caso de uso termina.	
Flujo Alternativo	
Título	**Descripción**
N/A	N/A
Precondiciones	
Título	**Descripción**
Existencia	Debe existir al menos una segmentación asociada a dicha página Web en el sistema.
Postcondiciones	
Título	**Descripción**
Éxito	El sistema devuelve la lista de segmentaciones que conforman el *ground truth* de la página Web.
Fracaso	El sistema devuelve un mensaje de error.

Tabla 15: Repositorio MoB UC.3.2.1.3

Nombre	**UC.3.2.1.3.1:** Eliminar una segmentación
Descripción	El usuario identificado como administrador puede eliminar una segmentación determinada.
Actor	Usuario Administrador
Flujo de Eventos	
Flujo Básico	
El caso de uso inicia cuando el usuario se encuentra consulta una lista de segmentaciones de una página Web. 1. El usuario selecciona la opción de eliminar una segmentación de la lista. 2. El sistema envía una alerta de confirmación. 3. El usuario confirma la acción. 4. El sistema elimina la segmentación seleccionada de la lista. 5. El caso de uso termina.	
Flujo Alternativo	
Título	**Descripción**
Cancelar	1. El usuario selecciona la opción de eliminar una segmentación de la lista. 2. El sistema envía una alerta de confirmación. 3. El usuario cancela la acción.
Precondiciones	
Título	**Descripción**
Usuario identificado	El usuario debe estar identificado en el sistema.
Usuario administrador	El usuario debe poseer el rol de administrador.
Postcondiciones	
Título	**Descripción**
Éxito	La segmentación es eliminada del sistema.
Fracaso	La segmentación no pudo ser eliminada por alguna razón.

Tabla 16: Repositorio MoB UC.3.2.1.3.1

Nombre	**UC.4:** Consultar documentación de MoB API
Descripción	El usuario puede consultar la descripción de los diferentes servicios ofrecidos por el API de MoB y la

	forma de usarlos.
Actor	Usuario y Usuario Administrador
Flujo de Eventos	
Flujo Básico	
El caso de uso inicia apenas el usuario hace clic sobre “MoB API doc”. 1. El sistema presenta un documento con la descripción de todos los servicios ofrecidos por el API de MoB y la forma de utilizar cada uno. 2. El caso de uso termina.	
Flujo Alternativo	
Título	**Descripción**
N/A	N/A
Precondiciones	
Título	**Descripción**
N/A	N/A
Postcondiciones	
Título	**Descripción**
Éxito	El sistema devuelve la documentación del API de MoB.
Fracaso	El sistema devuelve un mensaje de error.

Tabla 17: Repositorio MoB UC.4

Nombre	**UC.5:** Consultar página de tutoriales
Descripción	El usuario puede visualizar una sección de ayuda con diferentes tips y guías para ayudar en su segmentación manual.
Actor	Usuario y Usuario Administrador
Flujo de Eventos	
Flujo Básico	
El caso de uso inicia apenas el usuario hace clic sobre “tutoriales”. 1. El sistema presenta una sección de guías y ayudas sobre la segmentación manual con la extensión de MoB. 2. El caso de uso termina.	

Flujo Alternativo	
Título	**Descripción**
N/A	N/A
Precondiciones	
Título	**Descripción**
N/A	N/A
Postcondiciones	
Título	**Descripción**
Éxito	El sistema devuelve una sección de instructivos referentes a la segmentación manual con la extensión MoB.
Fracaso	El sistema devuelve un mensaje de error.

Tabla 18: Repositorio MoB UC.5

Nombre	**UC.6:** Consultar página de descargas
Descripción	El usuario puede visualizar la sección de descargas donde se encuentran varios vínculos de interés.
Actor	Usuario y Usuario Administrador
Flujo de Eventos	
Flujo Básico	
El caso de uso inicia apenas el usuario hace clic en “descargas”. 1. El sistema presenta una sección con los vínculos para descargar la extensión MoB y el proyecto del sistema MoB. 2. El caso de uso termina.	
Flujo Alternativo	
Título	**Descripción**
N/A	N/A
Precondiciones	
Título	**Descripción**
N/A	N/A
Postcondiciones	

Título	Descripción
Éxito	El sistema devuelve la sección de descargas.
Fracaso	El sistema devuelve un mensaje de error.

Tabla 19: Repositorio MoB UC.6

Nombre	**UC.7:** Consultar lista de usuarios
Descripción	El usuario puede visualizar la lista de todos los usuarios registrados en el sistema.
Actor	Usuario y Usuario Administrador
Flujo de Eventos	
Flujo Básico	
El caso de uso inicia apenas el usuario hace clic sobre “lista de usuarios”. 1. El sistema presenta una lista de todos los usuarios registrados en el sistema. 2. El caso de uso termina.	
Flujo Alternativo	
Título	**Descripción**
N/A	N/A
Precondiciones	
Título	**Descripción**
Existencia	Debe existir al menos un usuario registrado en el sistema.
Postcondiciones	
Título	**Descripción**
Éxito	El sistema devuelve una lista de los usuarios registrados en el sistema.
Fracaso	El sistema devuelve un mensaje de error.

Tabla 20: Repositorio MoB UC.7

Nombre	**UC.7.1:** Consultar perfil de usuario
Descripción	El usuario puede consultar la información de cualquier usuario registrado.

Actor	Usuario y Usuario Administrador
Flujo de Eventos	
Flujo Básico	
El caso de uso inicia apenas el usuario realiza una búsqueda de un usuario en específico o selecciona uno de la lista de usuarios. 3. El sistema presenta la información del usuario registrado, junto con un vínculo hacia las segmentaciones realizadas por dicho usuario. 4. El caso de uso termina.	
Flujo Alternativo	
Título	**Descripción**
N/A	N/A
Precondiciones	
Título	**Descripción**
Existencia	El usuario cuyo perfil se quiere observar, debe estar registrado en el sistema.
Postcondiciones	
Título	**Descripción**
Éxito	El sistema devuelve la información del usuario.
Fracaso	El sistema devuelve un mensaje de error.

Tabla 21: Repositorio MoB UC.7.1

Nombre	**UC.7.1.1:** Cambiar rol de usuario
Descripción	El usuario identificado como administrador puede modificar el rol de otros usuarios.
Actor	Usuario Administrador
Flujo de Eventos	
Flujo Básico	
El caso de uso inicia cuando el usuario administrador consulta la información del usuario que va a ser modificado. 1. El sistema presenta una lista desplegable de roles en la tabla de información del usuario. 2. El usuario administrador selecciona un rol de la lista y presiona “guardar”.	

3. El sistema actualiza el rol para el usuario. 4. El caso de uso termina.	
Flujo Alternativo	
Título	**Descripción**
N/A	N/A
Precondiciones	
Título	**Descripción**
Usuario identificado	El usuario debe estar identificado en el sistema.
Usuario administrador	El usuario debe poseer el rol de administrador.
Postcondiciones	
Título	**Descripción**
Éxito	El rol del usuario es actualizado.
Fracaso	El rol del usuario no puede ser actualizado por alguna razón.

Tabla 22: Repositorio MoB UC.7.1.1

Nombre	**UC.8:** Consultar información de las etiquetas
Descripción	El usuario puede consultar la descripción de todas las etiquetas presentes en el sistema.
Actor	Usuario y Usuario Administrador
Flujo de Eventos	
Flujo Básico	
El caso de uso inicia apenas el usuario hace clic sobre “etiquetas”. 1. El sistema presenta una lista de etiquetas,cada una con su respectiva descripción. 2. El caso de uso termina.	
Flujo Alternativo	
Título	**Descripción**
N/A	N/A
Precondiciones	
Título	**Descripción**

Existencia	Debe existir al menos una etiqueta en el sistema.
Postcondiciones	
Título	**Descripción**
Éxito	El sistema devuelve una lista de etiquetas.
Fracaso	El sistema devuelve un mensaje de error.

Tabla 23: Repositorio MoB UC.8

Nombre	**UC.8.1:** Administrar etiquetas
Descripción	El usuario identificado con rol de administrador puede realizar diversas tareas de administración (editar, eliminar, crear) sobre las etiquetas.
Actor	Usuario Administrador
Flujo de Eventos	
Flujo Básico	
El caso de uso inicia cuando el usuario selecciona “Editar etiquetas” en la página de etiquetas. 5. El sistema presenta una tabla con todas las etiquetas existentes y las opciones de administrador. 6. El usuario selecciona una de las acciones de administrador. 7. El sistema aplica los cambios. 8. El caso de uso termina.	
Flujo Alternativo	
Título	**Descripción**
Crear etiqueta	5. El usuario escoge crear una etiqueta. 6. El sistema le presenta un formulario. 7. El usuario rellena el formulario y envía. 8. El sistema valida los datos y agrega la etiqueta a la lista.
Eliminar etiqueta	5. El usuario escoge eliminar una etiqueta. 6. El sistema le presenta un aviso de verificación 7. El usuario acepta. 8. El sistema elimina la etiqueta de la lista.
Editar etiqueta	5. El usuario escoge editar una etiqueta.

	6. El sistema le presenta un formulario completado con los datos preexistentes de la etiqueta. 7. El usuario edita los datos y envía. 8. El sistema valida los datos y actualiza la información en la lista.
Precondiciones	
Título	**Descripción**
Usuario identificado	El usuario debe estar identificado en el sistema.
Usuario administrador	El usuario debe poseer el rol de administrador.
Postcondiciones	
Título	**Descripción**
Éxito	La información de la lista de etiquetas se actualiza exitosamente de acuerdo al cambio realizado por el usuario.
Fracaso	La acción de administración no se realiza por alguna razón.

Tabla 24: Repositorio MoB UC.8.1

Nombre	**UC.9:** Cambiar de idioma
Descripción	El usuario puede cambiar el idioma de la extensión entre español, inglés o francés.
Actor	Usuario y Usuario Administrador
Flujo de Eventos	
Flujo Básico	
El caso de uso inicia cuando se activa el icono de la extensión. 5. El sistema presenta los tres posibles idiomas: español, inglés o francés. 6. El usuario selecciona uno de los tres idiomas ofrecidos. 7. El lenguaje del sistema cambia de acuerdo a la selección. 8. El caso de uso termina.	
Flujo Alternativo	
Título	**Descripción**
N/A	N/A

Precondiciones	
Título	**Descripción**
N/A	N/A
Postcondiciones	
Título	**Descripción**
Éxito	El lenguaje de la extensión se modifica según el lenguaje seleccionado.
Fracaso	El lenguaje no se modifica, se queda en inglés por defecto.

Tabla 25: Repositorio MoB UC.9

- **Colaboración:**

El desarrollo del sitio Web MoB Repository se desarrolló haciendo uso del lenguaje de marcado HTML5 para la estructura de las páginas, las reglas de estilo CSS3 (*c.f* Capítulo 2.1.8.1) para la apariencia de las mismas, y el lenguaje de programación Javascript junto con un framework del mismo llamado Jquery (*c.f* Capítulo 2.1.8.2) para el comportamiento de las páginas y control de eventos. Todo esto del lado del cliente, del lado del servidor, como se mencionó antes, está apoyado por el API de MoB (Python y Flask) y la base de datos conectada a éste (Postgresql y Postgis).

Debido a que la Tabla X anteriormente mostrada, explica las funcionalidades que posee el sitio, no se considera necesario extender dicha explicación aún más.

- **Aprendizaje:**

El desarrollo de este sitio no estaba contemplado en el trabajo original, pero se creyó pertinente poder tener una interfaz para que los usuarios pudiesen observar las segmentaciones realizadas y otro tipo de usuario (administradores) poder controlar estos datos de una forma más sencilla, por esta razón se desarrolló el MoB Repository, no se cumplen ninguno de los objetivos específicos originales, pero complementa de forma excelente los componentes principales de este trabajo.

4.3.4. 4to Ciclo: La Mejor Segmentación

En el desarrollo de este ciclo se busca crear una segmentación denominada “la mejor segmentación” de una página Web, a partir de la colección de todas segmentaciones realizadas sobre esa página Web, también conocido como *ground truth*.

- **Especulación:**

En la etapa de especulación de este ciclo se realizó el siguiente análisis con la finalidad de encontrar una solución a este determinado problema.

Objetivo:

Crear, a partir de un pool de segmentaciones (*ground truth*), una segmentación que represente la mejor segmentación de una página web.

Características de la mejor segmentación:

La mejor segmentación debe ser aquella que comparta la mayor cantidad de similitudes entre las demás segmentaciones, dado que, los usuarios pueden llegar a tener diferentes puntos de vista, pero los puntos de vistas que se comparten aseguran un acuerdo. Mientras más compartida sea dicha similitud, la definición de la misma se vuelve más verídica, al fin y al cabo la segmentación de la página es definida por cómo el usuario promedio la percibe.

Estructura del bloque:

Al llevar a cabo la comparación, los bloques recibirán una serie de puntajes dependiendo de sus similitudes geométricas y semánticas con respecto a los otros bloques, los puntajes de todos los bloques correspondientes a una misma segmentación podrán ser sumados dando la puntuación total de la segmentación, esto a su vez permite asegurar cierta fidelidad a la mejor segmentación a través de su puntaje, el cual no solo refleja si fue la mejor sino también la mejor entre cuantos (mientras mayor sea el puntaje obtenido, mayor es el número de segmentaciones que la respaldan). Varios de estos atributos se nombran durante el proceso de comparación, por lo que aquí se presenta una leyenda de los mismos:

- *Identificador:* Un identificador arbitrario del bloque.

- *Geometría:* La geometría del bloque (área, ubicación, entre otros)
- *Score Geométrico:* La puntuación obtenida por similitudes geométricas.
- *Etiqueta:* La etiqueta que le fue asignada al bloque.
- *Score Semántico:* La puntuación obtenida por similitudes semánticas.

Pasos en el proceso de comparación:

1. **Identificación de los bloques:** principalmente se deben identificar los bloques de la nueva segmentación a ser evaluada con respecto a los ya almacenados en la base de datos, puesto a que se quiere llevar un control de los bloques similares, es por esto que todos los bloques similares (geométricamente) llevan el mismo identificador arbitrario. Se toman los bloques de la nueva segmentación y se comparan por coincidencias entre todos los bloques de la base de datos (con un cierto margen de error).

2. **Contabilización de puntos:** para completar el paso anterior, se contabilizan todos los bloques bajo un mismo identificador, para obtener el score geométrico, después, dentro del mismo pool de bloques con el mismo identificador se contabilizan todos los que posean las mismas etiquetas, de esta forma obtener el score semántico.

3. **Creación de la mejor segmentación:** para crear la mejor mejor segmentación se incluyen todos aquellos bloques con identificadores pero cuyo score geométrico sea mayor que el 50% del número de segmentaciones realizadas, (esto garantiza que la mayoría opina que ese bloque debe existir). Dicho bloque poseerá la etiqueta más utilizada para ese bloque, es decir, se busca entre los de un mismo identificador la etiqueta que tenga el mayor score semántico.

- **Colaboración:**

El proceso de la creación de la mejor segmentación se lleva a cabo en el API, se hace uso del módulo “threading” y “queue” para lograr crear workers[11] los cuales realizarán el proceso de creación de la mejor segmentación en el servidor sin tener que hacer esperar al cliente.

[11] Los workers son hilos de procesos paralelos al proceso principal del sistema, se encargan de realizar tareas en el fondo sin interrumpir al proceso principal.

Para la primera etapa del proceso de creación de la mejor segmentación (**Identificación de los bloques**) se utiliza el siguiente query de Postgresql:

```
SELECT DISTINCT ON(g2.id_geo) ST_HausdorffDistance(g2.block,g.block) AS dist, g2.id_geo
AS id, g.id_geo, g.identifier
FROM mob_segmentation_geo AS g
INNER JOIN mob_segmentation_details AS d ON d.id_details = g.id_details
INNER JOIN mob_segmentation_geo AS g2 ON g.id_details <> g2.id_details
WHERE g2.id_details = id_segmentation
AND d.mob_status = 'scored'
AND d.mob_gran= gran
AND d.id_web_page = id_page
AND ST_HausdorffDistance(g2.block,g.block) < 30
ORDER BY g2.id_geo, dist, g.id_geo
```

Básicamente este query devuelve una lista con tres columnas: la primera conforma todos los ids de los bloques de la nueva segmentación sin repetirse, la segunda representa la distancia que separa el bloque del id anterior de algún otro bloque en el *ground truth* (siempre y cuando esta distancia sea menor que 30), la tercera columna representa el id de dicho bloque con el cual se comparó. Se debe aclarar que los ids anteriormente mencionado son los de la base de datos, no los identificadores, pero ahora que se conoce cuál bloque se asemeja a otro, se puede tomar el segundo bloque (que ya posee un identificador) y copiarlo al primero.

Cabe acotar que la función ST_HausdorffDistance[12] es usada gracias al componente de Postgis de Postgresql, y actúa sobre columnas del tipo Geométrico, las cuales cumplen un formato especial para interactuar con el componente.

En la segunda etapa del proceso de creación de la mejor segmentación (**Contabilización de puntos**) se utiliza los siguientes queries de Postgresql:

```
SELECT g.identifier, count(d.id_details) as score
FROM mob_segmentation_details as d
INNER JOIN mob_segmentation_geo as g ON g.id_details = d.id_details
WHERE mob_status = ‘scored’ AND id_web_page = id_page AND mob_gran = gran
GROUP BY g.identifier
ORDER BY g.identifier, score DESC
```

Este query devuelve una lista con dos columnas: la primera representa los identificadores de los bloques y la segunda indica una sumatoria de ids de bloques que

[12] La distancia Hausdorff es la mayor de todas las distancias existentes desde un punto en un conjunto hasta el punto más cercano en otro conjunto.

cumplen la condición de poseer el mismo identificador (viene representando el score geométrico).

```
SELECT g.identifier , g.tag, count(d.id_details) as score
FROM mob_segmentation_details as d
INNER JOIN mob_segmentation_geo as g ON g.id_details = d.id_details
WHERE mob_status = 'scored' AND id_web_page = id_page AND mob_gran = gran
GROUP BY g.identifier, g.tag
ORDER BY g.identifier, score DESC, g.tag
```

Este query es muy parecido al query anterior, con la diferencia de que ahora la agrupación del GROUP BY difiere pues se agrupan entre aquellos que poseen el mismo identificador Y la misma etiqueta (obteniendo el score semántico).

En la tercera etapa del proceso de creación de la mejor segmentación (**Creación de la mejor segmentación**) simplemente queda tomar todos los bloques que cumplan con la condición de que su identificador debe tener un score geométrico que sobrepase el 50% de las segmentaciones realizadas y su score semántico debe ser el mayor entre los de su mismo identificador.

- **Aprendizaje:**

Durante el desarrollo de este ciclo, se decidió hacer a un lado la idea original de elegir la mejor segmentación a partir de un proceso de validación social, pues esto requeriría de que los usuarios mismos votaran por la mejor segmentación, esto no resulta muy atrayente por las siguientes razones:

- Para que el proceso funcionase lo mejor posible, cada usuario que segmentase una página Web debería calificar todas las demás segmentaciones presentes para esa página Web, en el caso de ser pocas no habría muchos inconvenientes, pero en caso de ser más de 100 (que sería un caso ideal) representa un problema para el usuario, el cual tendrá que dedicar mucho tiempo en esta nueva tarea.

- Segmentar una página Web no es tarea sencilla, por lo que se quiere reducir el número de tareas del usuario lo más posible.

Por estos motivos se decidió crear la mejor segmentación sin intervención por parte de los usuarios, pero al mismo tiempo usando cada uno de sus puntos de vista.

Objetivos logrados:

- Desarrollar un sistema de recompensas y puntaje que vuelva interactivo y entretenido el proceso de segmentación.
- Proponer la mejor segmentación dentro de un grupo de usuarios que segmentan manualmente una misma página Web.

4.3.5. 5to Ciclo: Mejoras

En la realización de este ciclo se buscó optimizar los componentes del sistema, eliminar posibles complicaciones y mejorar el diseño de las interfaces.

- **Especulación:**

En esta etapa se analizó cómo se podría mejorar la experiencia del usuario, se concluyó que mantener un mismo diseño a lo largo de todo el sistema es una buena forma de no confundir al usuario.

- **Colaboración:**

Se eliminó el sistema de alertas que ofrece el navegador por defecto y se suplanta por uno creado personalizadamente, en la figura 29 se muestra la diferencia entre los dos.

Figura 29: Diferencia de Alertas

Se cambia el manejo de unidades de px (pixeles) a mm (milímetros) para que sea más comprensible para el usuario y no ocupe tanto espacio en las ventanas de información.Se actualizan las interfaces de la herramienta MoB y el MoB Repository para que resulten llamativas pero manteniendo su simplicidad. En la figura 30 y figura 31 se muestran ambas interfaces.

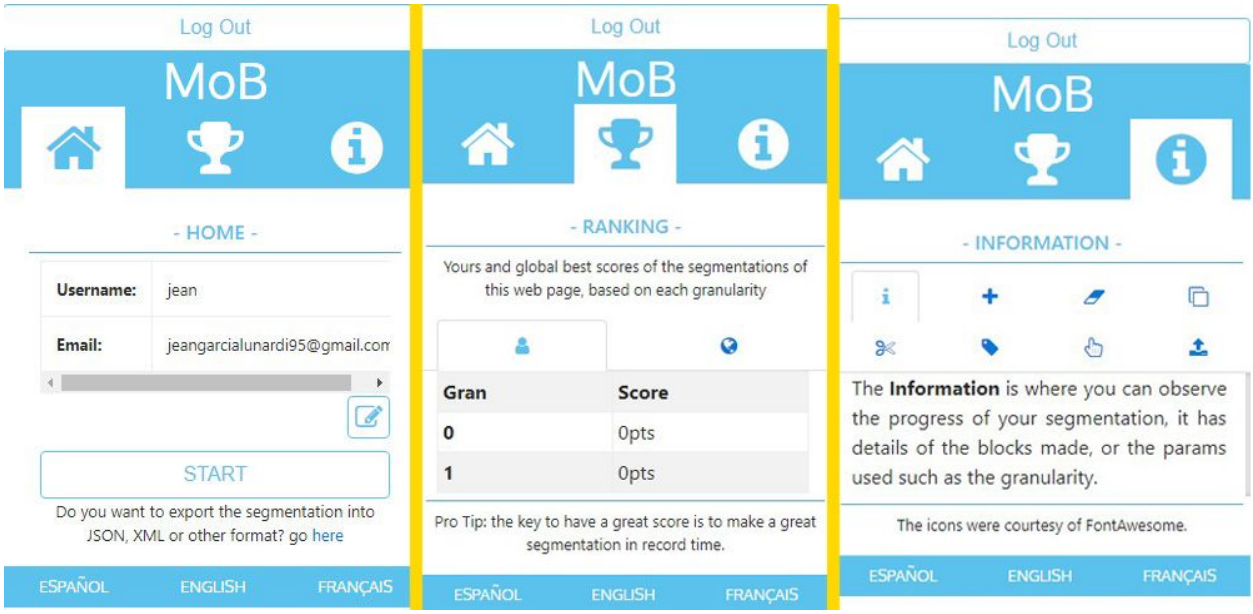

Figura 30: Interfaz Extensión MoB

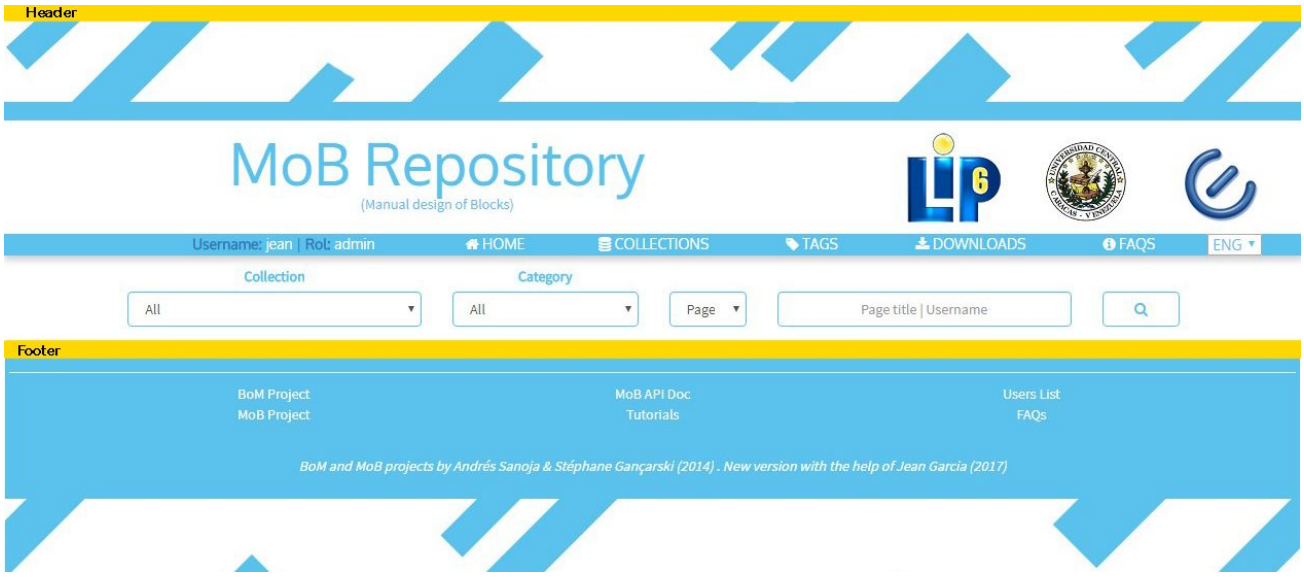

Figura 31: Interfaz Repositorio de MoB

- **Aprendizaje:**

Con la realización de este ciclo se espera llevar la experiencia del usuario a un nuevo nivel, ya que mientras más agradable resulte su experiencia con la herramienta aumentan las posibilidades de que realice más segmentaciones en diferentes páginas, y mientras más segmentaciones se obtengan, mejor será la *ground truth* del sistema y por consiguiente la mejor segmentación resultante.

Objetivos logrados:

- Mejorar las vistas de la herramienta para que esta herramienta cumpla con los requerimientos de IHC y sea una herramienta usable.

4.4. Pruebas de aceptación

Según el International Software Testing Qualification Board (ISTQB), las pruebas de aceptación son aquellas pruebas realizadas para determinar si un sistema satisface los criterios de aceptación que permitan que el usuario, cliente u otra entidad autorizada pueda determinar si acepta o no el sistema. A continuación se presentan dos tipos de pruebas realizadas para comprobar la aceptación de este sistema.

4.4.1. Pruebas funcionales

Para comprobar la funcionalidad del sistema, se realiza una prueba de caja negra. La cual consiste en comprobar si el sistema se comporta como es esperado en cada una de sus funcionalidades, no interesa la lógica de los procesos, únicamente los datos de entrada y salida (o el estado del sistema inicial y final), para ese fin, las siguientes tablas: tabla 26 y tabla 27, se apoyan en los casos de usos descritos anteriormente, así como en las descripciones de las acciones de la herramienta de segmentación manual MoB, para establecer si se cumple o no el comportamiento esperado.

Caso de Uso	**Se realiza el comportamiento esperado**
UC.1: Consultar información sobre las funcionalidades de la herramienta.	Verdad
UC.2: Cambiar idioma.	Verdad
UC.3: Consultar puntuaciones de la segmentación de la página.	Verdad
UC.4: Registrarse en el sistema.	Verdad
UC.5: Identificarse en el sistema.	Verdad
UC.6 Realizar segmentación Manual.	Verdad
UC.6.1: Insertar bloque.	Verdad
UC.6.2: Eliminar bloque.	Verdad
UC.6.3: Unir bloques.	Verdad
UC.6.4: Cortar bloques.	Verdad
UC.6.5: Etiquetar bloque.	Verdad
UC.6.6: Seleccionar bloque.	Verdad
UC.6.7: Panel de información.	Verdad

UC.6.8: Enviar segmentación.	Verdad

Tabla 26: Prueba Caja Negra de Extensión MoB

En la tabla 26 se indica que durante las pruebas de caja negra de la extensión se cumplieron todos los comportamientos esperados.

Caso de Uso	**Se realiza el comportamiento esperado**
UC.1: Visualizar página principal.	Verdad
UC.2: Realizar una búsqueda.	Verdad
UC.3: Consultar colecciones.	Verdad
UC.3.1: Administrar colecciones.	Verdad
UC.3.2: Consultar categorías.	Verdad
UC.3.2.1: Consultar información de una página Web.	Verdad
UC.3.2.1.1: Descargar información de una segmentación o página Web.	Verdad
UC.3.2.1.2: Eliminar página Web.	Verdad
UC.3.2.1.3: Consultar lista de segmentaciones de una página Web.	Verdad
UC.3.2.1.3.1: Eliminar segmentación.	Verdad
UC.4: Consultar documentación de MoB API.	Verdad
UC.5: Consultar página de tutoriales.	Verdad
UC.6: Consultar página de descargas.	Verdad
UC.7: Consultar lista de usuarios.	Verdad
UC.7.1: Consultar perfil de usuario.	Verdad
UC.7.1.1: Cambiar rol de usuario.	Verdad
UC.8: Consultar información de las etiquetas.	Verdad
UC.8.1: Administrar etiquetas.	Verdad

UC.9: Cambiar idioma.	Verdad

Tabla 27: Prueba Caja Negra de Repositorio MoB

En la tabla 27 se observa que durante la prueba de caja negra, todos los comportamientos esperados por parte del Repositorio de MoB se llevan a cabo exitosamente.

4.4.2. Pruebas no funcionales

Para las pruebas no funcionales se quiere medir la usabilidad del sistema, específicamente de la herramienta de segmentación manual MoB, para esto se evalúa la reacción se 5 individuos ante el sistema. Se observó las reacciones de los participantes mientras completan 2 objetivos planteados y finalmente se les dió un cuestionario para que responder según su experiencia. Cabe acotar que la herramienta está orientada a usuarios de 13 años en adelante, con inexistentes, bajos, intermedio o avanzados conocimientos en segmentación de páginas Web.

Objetivos:

1.- Realizar una segmentación manual sobre la página Web: https://wiki.apache.org/httpd/RedirectSSL.

2.- Visitar el Repositorio de MoB y observar la segmentación realizada.

A continuación se presentan los resultados del cuestionario realizado.

Seccion General:

En esta sección se hicieron preguntas de los objetivos en general y el sistema en general.

Figura 32: Pregunta 1 del Cuestionario

En la figura 32 se puede observar que el 60% de los participantes respondieron tener un conocimiento “intermedio” en cuanto a la segmentación de páginas Web. 20% un conocimiento “bajo” y un 20% un conocimiento “inexistente”, es decir, inicialmente los participantes no son expertos en el área de segmentación Web, esto quiere decir que los resultados nos indicarán si el sistema es usable para personas inexpertas en el área (lo cual sería lo ideal).

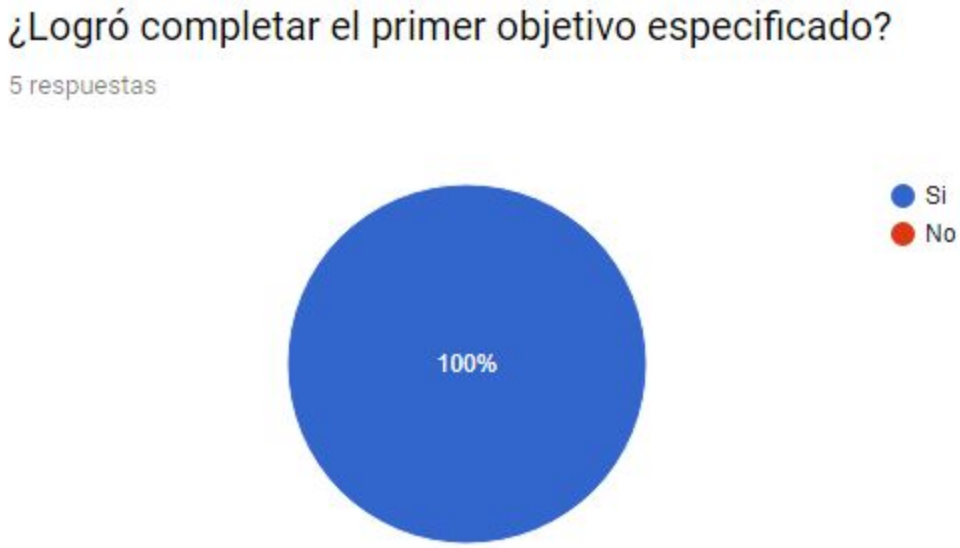

Figura 33: Pregunta 2 del Cuestionario

La figura 33 indica que efectivamente el 100% los participantes pudieron segmentar la página Web y enviar sus resultados.

Figura 34: Pregunta 3 del Cuestionario

En la figura 34 se observa que el 60% de los participantes tomó entre 5 a 10 minutos, 20% entre 10 a 20 minutos y 20% entre 0 a 5 minutos. Las respuestas obtenidas más la experiencia comentada por los participantes durante la segmentación se concluye que la herramienta permite una rápida edición de los bloques.

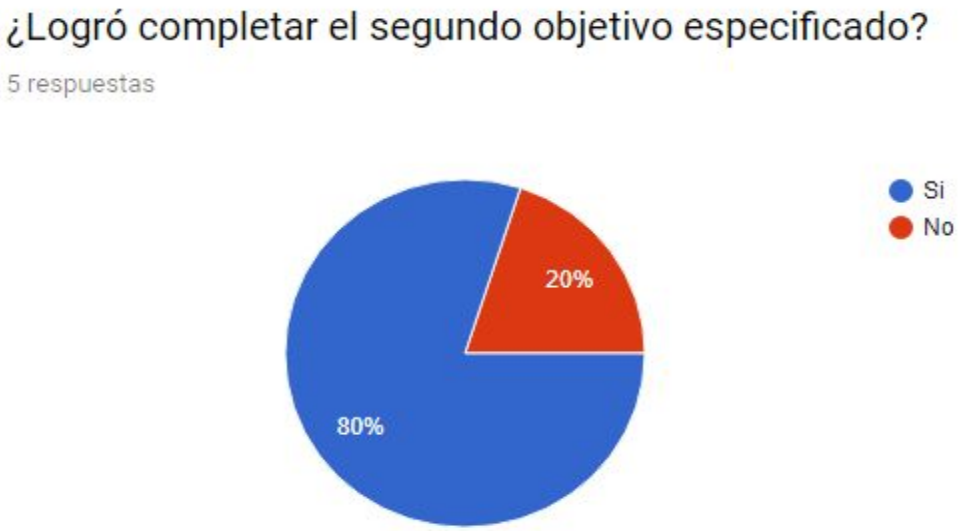

Figura 35: Pregunta 4 del Cuestionario

En la figura 35 se muestra que el 80% de los participantes lograron encontrar su propia segmentación en el Repositorio de MoB mientras que un 20% no lo logró. Se concluye, según los resultados obtenidos y las reacciones observadas de los participantes, que la mayoría de los usuarios confunde a primera vista la página de información de la página Web como la página de información de su propia segmentación, sin embargo la mayoría logra encontrar finalmente su propia segmentación de una forma u otra.

Figura 36: Pregunta 5 del Cuestionario

En la figura 36 se evidencia que el 60% de los participantes tardaron entre 0 a 5 minutos en encontrar su propia segmentación y el 40% de 5 a 10 minutos. Cabe acotar que la persona que no logró completar el segundo objetivo tomo de 5 a 10 minutos antes de rendirse, esto es una información importante, pues ciertamente el usuario después de un tiempo de no lograr conseguir lo que busca se frustra y se rinde, por suerte la mayoría de las personas logró completar el objetivo de forma rápida, por lo que se puede concluir que a pesar de que la navegación puede resultar un poco confusa, la mayoría logra comprender rápidamente la lógica de la misma.

Sección de MoB Extensión:

En esta sección se realizaron preguntas con respecto a la extensión de MoB.

Figura 37: Pregunta 6 del Cuestionario

En la figura 37 se puede observar que el 100% de los usuarios considera que el esquema de colores elegido es agradable. Cabe acotar que durante la observación los participantes comentaron sobre el agradable estilo de la extensión y la herramienta de segmentación.

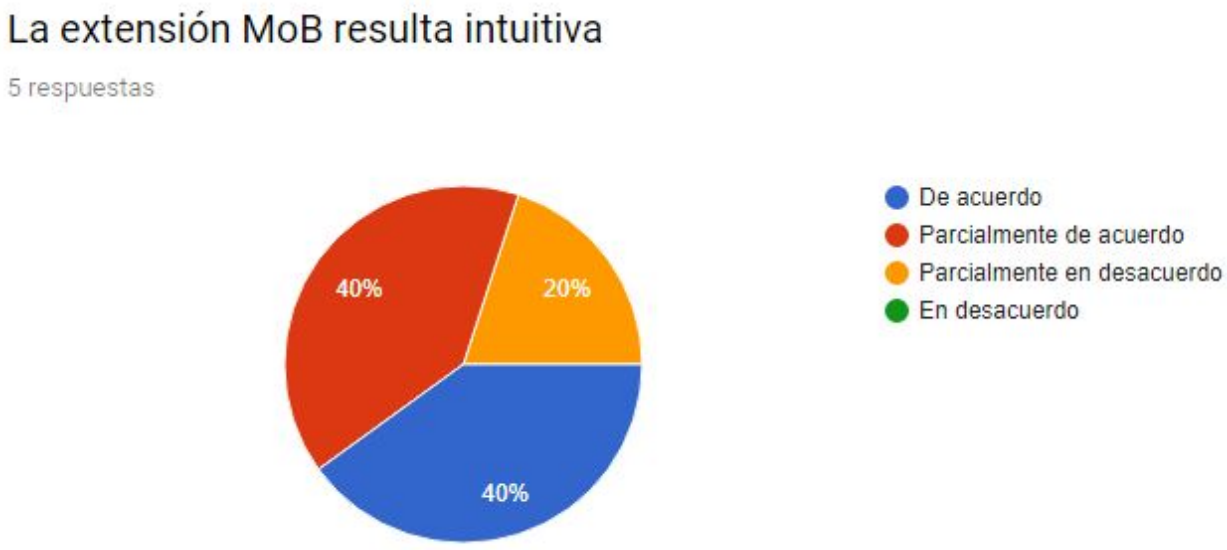

Figura 38: Pregunta 7 del Cuestionario

En la figura 38 se evidencia que existe cierta discordancia respecto a la intuitividad de la extensión, al 40% de los participantes considera que la herramienta es

intuitiva, al 40% les parece poco intuitiva y un 20% no la considera muy intuitiva, es decir, el 60% (la mayoría) considera que la extensión es poco intuitiva.

Figura 39: Pregunta 8 del Cuestionario

Según los resultados observados en la figura 39, un 60% considera que las acciones de la herramienta de segmentación son relativamente fáciles de usar, y un 40% considera que son efectivamente fáciles de usar. Se concluye que la mayoría de los participantes no parecen convencidos de la facilidad de usar las acciones, algunos de los comentarios dado por los participantes durante la prueba demuestra que efectivamente saben para qué sirve la acción pero no están seguros de cómo usarla, por ejemplo la acción “cortar bloques”, no les resulta evidente a simple vista que se debe seleccionar dos bloques y que el primero se conserva y el segundo se corta, estos conocimientos llegan a ellos una vez repasan la sección de información de la extensión.

Figura 40: Pregunta 9 del Cuestionario

En la figura 40 se muestra que un 80% terminó comprendiendo la finalidad y la forma de emplear todas las acciones de las herramientas, mientras un 20% no, esto demuestra que hay acciones que presentan un nivel intermedio en complejidad y requieren de una breve instrucción previa.

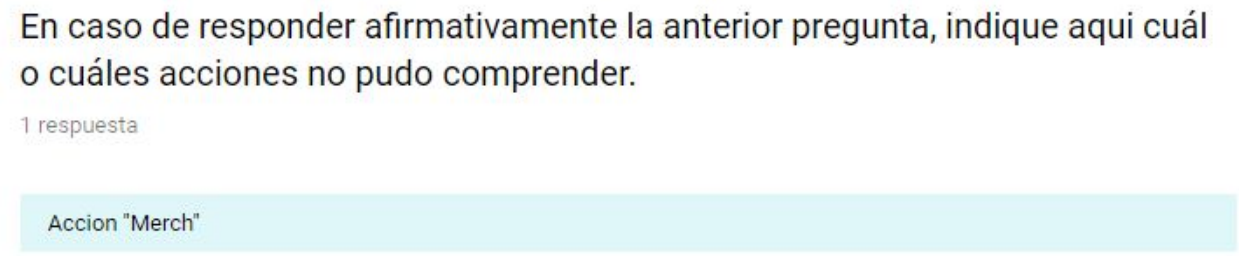

Figura 41: Pregunta 10 del Cuestionario

La figura 41 muestra una pregunta complementaria a la anterior, el participante que no entendió por completo todas las acciones de la herramienta indica el nombre de la acción que no pudo comprender: “merch” refiriéndose a la acción “merge” o “unión de bloques”.

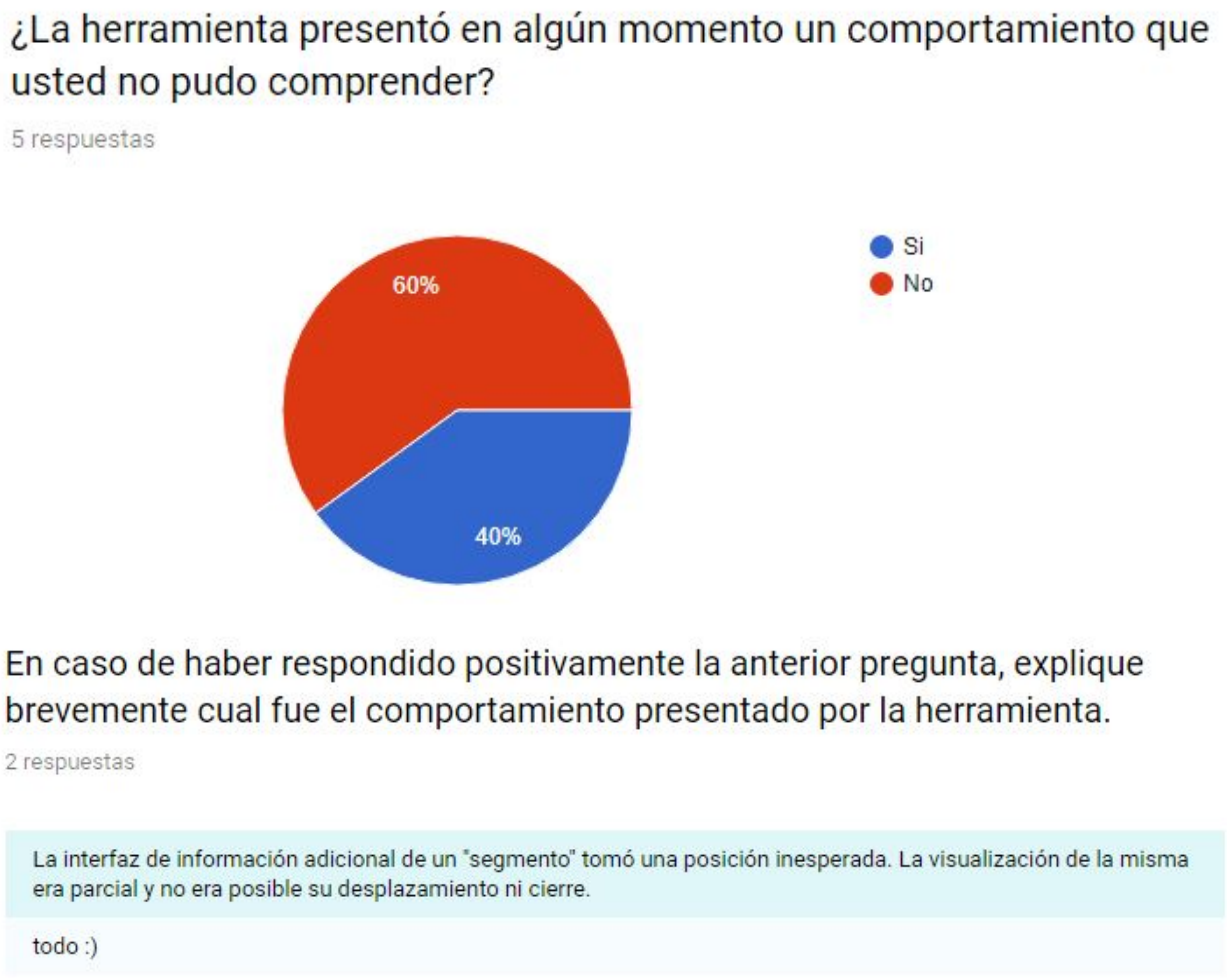

Figura 42: Pregunta 11 del Cuestionario

En la figura 42 se muestran dos preguntas complementarias, En la primera, un 60% de los participantes afirman no haber tenido ningún problema con el comportamiento de la herramienta, mientras que un 40% afirma lo contrario. Analizando las respuestas de la segunda pregunta, se observa que uno de los participantes encontró un “bug” en la herramienta donde el cuadro de información del bloque queda parcialmente oculto y no era posible su cierre. El segundo participante usa una respuesta ambigua “todo :) “ la cual no es posible evaluar, salvo el hecho de que quedó insatisfecho con el uso de la herramienta.

Sección de MoB Repository:

En esta sección se realizan las preguntas referentes al Repositorio MoB.

Figura 43: Pregunta 12 del Cuestionario

En la figura 43 se demuestra nuevamente que a los participantes les agrada el diseño y esquema de colores de la interfaz del sistema, ambas interfaces tienen el mismo diseño y esquema de colores, esta pregunta sirve para certificar que, en cuanto al aspecto estético, el sistema MoB es aceptado por los participantes.

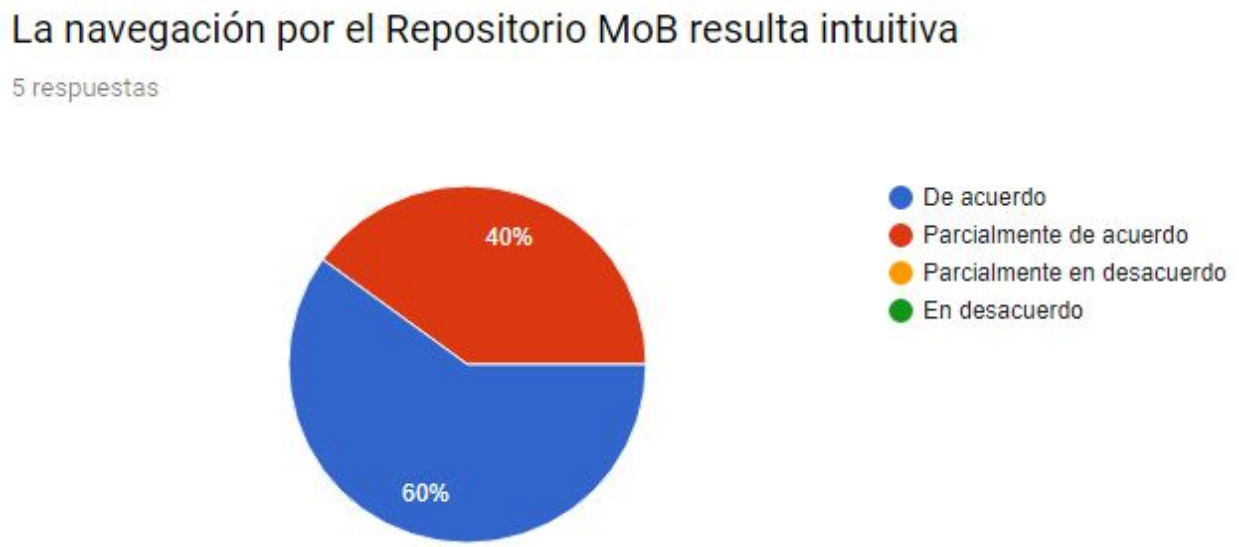

Figura 44: Pregunta 13 del Cuestionario

En la figura 44 se evidencia que un 60% de los participantes considera que la navegación en el Repositorio es intuitiva, mientras que un 40% la considera poco intuitiva. Esto aunado a las observaciones realizadas, se concluye que efectivamente se requiere de un poco de tiempo antes de que el usuario encuentre la lógica de navegación del sistema.

Figura 45: Pregunta 14 del Cuestionario

En la figura 45 se presenta una declaración muy parecida a la anterior, en la figura 44. Sin embargo la presente trata específicamente sobre encontrar la página de información de la segmentación que realizó el participante. un 80% (los mismos participantes que lograron completar el segundo objetivo) indicaron que se les hizo relativamente fácil encontrar su segmentación (según las respuestas de la figura 36, de 0 a 10 minutos). Mientras que el participante que no logró completar el objetivo indica que no le resultó sencillo (20%).

En conclusiones generales, basándose en las respuestas obtenidas del cuestionario, los comentarios hechos por los participantes y el comportamiento observado de los mismos, se tiene que:

- El sistema en general presenta un aspecto estético agradable para los usuarios.

- La herramienta de segmentación permite a los usuarios inexpertos realizar segmentaciones rápidamente sobre una página Web, de una forma sencilla.
- Algunas de las acciones dentro de la herramienta de segmentación requieren de una breve instrucción previa.
- La navegación general del Repositorio MoB es entendible, sin embargo cuando se debe profundizar, como por ejemplo, buscar segmentaciones específicas, el usuario debe invertir un poco de tiempo en entender la lógica de la navegación.
- Se puede considerar que el sistema MoB es usable, sin embargo, dada la complejidad del mismo, requiere de un breve periodo de aprendizaje por parte del usuario.
- Dada la inexperiencia de la mayoría de los participantes en el tema de segmentaciones de páginas Web, dichas segmentaciones terminaron siendo muy diferentes entre ellas, compartiendo casi ninguna característica similar, resultando en que la mejor segmentación estuviese prácticamente vacía, esto no es precisamente erroneo, pues son solo 5 segmentaciones, al ser 50 claramente habrían más características compartidas, sin embargo, esto es un indicador de que se debe alentar al usuario a consultar la sección de tutoriales y guías dentro del Repositorio, sobre todo si es un segmentador principiante.

Nota: Cabe aclarar que este cuestionario no tenía la intención de medir el correcto funcionamiento del sistema (para eso se hizo las pruebas funcionales), por esa razón se pasó por alto el “bug” presentado en uno de los participantes.

CAPÍTULO V: CONCLUSIONES

A pesar de ser un proyecto que se apoya en investigaciones pasadas, no significa que el desarrollo del mismo se llevó a cabo sin ningún problema. Durante el desarrollo del presente trabajo se presentaron diferentes desafíos, a continuación se hace una breve descripción de los desafíos presentados en cada componente del sistema y la conclusión obtenida de cada uno.

En cuanto a la estrella principal del sistema, la herramienta de segmentación manual (Extensión de MoB), presentó uno de los retos más grandes del mismo. Se requirió no solo que cumpliese con su misión de segmentar la página Web, lo cual ya de por sí es retador cuando el mundo del diseño de las páginas Web se encuentra en constante cambios, pero además se requería presentarle al usuario una herramienta de fácil uso. Por ejemplo, que al alcance de un clic pudiese ser capaz de recortar una sección de la página Web, que pudiese editar la sección creada de diferentes formas e incluso recortar posibles secciones que se intersectan, entre otras funcionalidades, todo de forma cómoda, rápida y entendible para el usuario. Al cabo de las pruebas de caja negra y usabilidad se dio a conocer que el objetivo fue logrado, aunque el usuario debe pasar por un breve periodo de instrucción previo ofrecido por el mismo sistema donde se le explica la tarea que debe realizar, esto se debe a la complejidad inherente que presenta la tarea de realizar una segmentación manual sobre una página Web.

En cuanto al repositorio de MoB, el mayor reto fue la forma de encontrar una forma comprensible de ordenar toda la información sobre las páginas Web y sus segmentaciones, después de las pruebas de caja negra se evidencio que el repositorio funcionaba como se esperaba, pero en las pruebas de usabilidad se evidenció cierta dificultad por parte del usuario al momento de navegar por el repositorio de MoB, requiere que el usuario lea las indicaciones presentadas por el sistema para entender la lógica del mismo. Esta información recopilada puede ser tomada en cuenta para las mejoras a futuro que se vayan a realizar sobre el sistema.

Entre los retos presentados en el desarrollo del API de MoB se encontraba: el poder desarrollar todos los servicios pertinentes de la forma más modular posible, presentando estructuras de respuesta que fuesen fácil de comprender y manejar, es especial la estructura de bloques que se debe pasar para cargar los datos de la segmentación, también presentó un reto el análisis de la mejor segmentación, dicho proceso fue desarrollado para ser ejecutado en el API de MoB, como un hilo paralelo

para no mantener esperando al cliente, este era uno de los retos presentados para el análisis, ya que se debe realizar dicho análisis cada vez que se carga una nueva segmentación en el sistema. Otro de los retos fue la forma de comparar los diferentes bloques de forma rápida y eficaz, gracias a Postgis con sus objetos Geométricos y funciones asociadas a estos, este reto pudo ser superado sin muchos inconvenientes.

Este TEG comenzó con la idea de desarrollar una herramienta para ayudar en el desarrollo de otros proyectos. Sin embargo, a lo largo del desarrollo se fueron extendiendo las funcionalidades principales y añadiendo funcionalidades adicionales a los elementos principales del sistema, reforzando la funcionalidad y permitiendo la evolución del mismo. Como resultado tenemos un sistema bastante completo en donde se pueden realizar segmentaciones manuales, incluir segmentaciones hechas por algoritmos y ser almacenadas. La información de las segmentaciones puede ser mostrada a través de una interfaz Web (Repositorio MoB), y en el caso de las segmentaciones manuales, pueden ser analizadas y obtener ese elemento que representa la razón principal por la cual se realiza este trabajo investigativo, la mejor segmentación.

Se considera que en la realización de este trabajo se completaron exitosamente los objetivos planteados e incluso se dio un paso extra, sin embargo esto no significa que este trabajo representa la solución definitiva para la problemática expuesta, se espera que el presente trabajo especial pueda servir de base o inspiración para más trabajos relacionados que experimenten con otros enfoques.

1. Contribución

Como se ha mencionado a lo largo de esta investigación, el presente trabajo representa en sí un elemento muy importante en un sistema mucho más grande. El resultado del sistema desarrollado es aquella segmentación llamada “mejor segmentación”. Esta segmentación representa la segmentación que será usada como parámetro de evaluación para los algoritmos de segmentación.

Se puede resumir que la contribución del presente trabajo es el ofrecer una forma de evaluación para poder calificar la calidad de los algoritmos de segmentación para que los mismos se puedan encaminar más a la forma de segmentación hecha por un usuario.

2. Recomendaciones

Si se desea hacer uso del sistema como base para el desarrollo de otro o realizar modificaciones en el mismo, es recomendado fuertemente que se consulte las instrucciones de instalación (*c.f* Anexos 1).

Para hacer uso de los servicios ofrecidos por la API RESTful se recomienda leer la documentación del MoB API (*c.f* Anexos 2)

3. Trabajos Futuros

La inclusión de otros formatos de exportación para las segmentaciones sería una buena actualización.

Según los resultados arrojados por la prueba de usabilidad, el Repositorio MoB presenta una navegación que puede resultar algo confusa para algunos usuarios, es por eso que se propone como un trabajo a futuro la implementación de una navegación más intuitiva.

Además, es importante destacar que la extensión actual funciona únicamente para el navegador Chrome/Chromium es por eso que se plantea un trabajo a futuro donde se adapte dicha extensión para funcionar en una mayor variedad de navegadores (Firefox, Safari, Opera, entre otros).

ANEXOS

1. Instalación del Ambiente

A continuación se describen los pasos para la configuración del ambiente de desarrollo que se debe tener para correr el sistema. Los archivos se pueden encontrar en el repositorio:

1.1. Configuración del Servidor

Necesitarás de Python 3.5 o superior y módulos extras.

I. Instalar Python 3.5:

- Ubuntu:
Si posees Ubuntu 16 o superior probablemente ya tu sistema posea Python 3.5, sin embargo el comando sería:

```
sudo apt-get install python3
```

- Windows:
Puedes descargar el instalador directamente de su página oficial: <https://www.python.org/downloads/windows/>

II. Instalar los Módulos

Puedes instalar cada módulo usando la herramienta **pip** que viene con la instalación de Python 3. Al lado de cada nombre se especifica el comando de instalación.

- flask (0.12.2) -> “sudo pip3 install flask”
- flask_cors (3.0.3) -> “sudo pip3 install flask_cors”
- flask_mail (0.9.1) -> “sudo pip3 install flask_mail”
- psycopg2 (2.7.3) -> “sudo pip3 install psycopg2”
- bs4 (0.0.1) -> `sudo pip3 install bs4`

1.2. Para el Sistema Manejador de Base de Datos

I. Instalar la base de datos requerida, Postgresql (v.10.1):

- Ubuntu:

```
sudo apt-get install postgresql-10.1
```

- Windows:
Puede descargar el instalador directamente de su página oficial: https://www.postgresql.org/download/

II. Instalar el complemento de Postgis (v.2.4):

- Ubuntu:

```
sudo apt install postgis postgresql-10-postgis-4
```

y

```
sudo -u postgres psql -c "CREATE EXTENSION postgis; CREATE EXTENSION postgis_topology;" gisdata
```

- Windows:
(Debe usar Windows de 64 bits)

1. Puede elegirlo de la lista de complementos/extensiones que le ofrece PGAdmin 4 (Viene con la instalación de Postgresql).

2. Puede instalar los binarios desde: https://postgis.net/windows_downloads/

Finalmente, se requiere la importación de los modelos para la base de datos, dichos modelos se encuentran en el archivo *"MoB-API-and-MoB-Repository/db_models_sql.sql"* en el repositorio de Github, con usar esos queries en Postgresql debería ser suficiente para crear los modelos necesarios, también puede "restaurar" usando la versión **tar** del archivo en *"MoB-API-and-MoB-Repository/db_models_tar"*.

1.3. Últimos ajustes

Se deben modificar las siguientes líneas en los archivos que se indican, para reemplazarlos con la información apropiada de quien desarrolla el proyecto.

MoB-API-and-MoB-Repository/mob_api.py (línea 20)

```
conn = psycopg2.connect(dbname='db_name', user='postgres_user',
password='mypassword', port='5432')
```

Se debe colocar la información de la base de datos a ser usada.

MoB-API-and-MoB-Repository/mob_api.py (línea 31, 32)

```
app.config['MAIL_USERNAME'] = 'myemail@gmail.com'
app.config['MAIL_PASSWORD'] = 'password'
```

Se debe colocar el correo y contraseña del correo del que se desee que se envíen los anuncios y mensajes de MoB.

MoB-API-and-MoB-Repository/mob_api.py (línea 39, 40, 41)

```
api_url = 'https://mob.ciens.ucv.ve/'
mobEnvURL = '/route-to-warcs-files/'
hash_key = 'jean'
```

Se debe colocar la url del dominio donde se correrá el sistema, la palabra clave para la codificación hash y la URL absoluta de donde se desea almacenar los archivos WARC que se crearán, los archivos WARC son almacenados en la siguiente ruta:

```
ruta = mobEnvURL+str(id)
```

MoB-Extension/js/script.js (línea 18)

```
var api_url = 'https://mob.ciens.ucv.ve';
```

Se debe colocar la url del dominio.

MoB-Extension/js/popup.js (línea 4)

```
var api_url = 'https://mob.ciens.ucv.ve';
```

Se debe colocar la url del dominio.

Nota: Se debe tener en cuenta que si se usa un sistema operativo Windows, la funcionalidad que permite descargar las páginas en formato WARC no funcionará, pues "wget" es una utilidad que ya viene instalada en los sistemas Ubuntu/Linux, para usarlo en Windows se deberá instalar manualmente y quizás modificar la línea de ejecución :

Instalarlo desde: http://gnuwin32.sourceforge.net/packages/wget.htm

MoB-API-and-MoB-Repository/mob_api.py (línea 258)

```
subprocess.run('wget --no-check-certificate
--warc-file='+mobEnvURL+str(id)+' --recursive --level=1 -O tempfile
'+url, shell=True)
```

Se debe modificar la línea de comando dependiendo de la forma en que se haya instalado el wget o si se encuentra entre las variables del sistema.

2. Documentación de MoB API

A continuación se presenta la sintaxis adecuada para solicitar los diferentes servicios de la API de MoB, junto con la lista de todos los posibles servicios y qué esperar de cada uno.

Lista de servicios:

A. Registrar usuario.
B. Iniciar sesión.
C. Cerrar sesión.
D. Recuperar contraseña.
E. Cargar segmentación.
F. Obtener vista previa de segmentación.
G. Obtener lista de colecciones.
H. Obtener lista de etiquetas.
I. Obtener lista de puntajes globales.
J. Obtener lista de puntajes individuales.
K. Obtener página Web en formato WARC.
L. Obtener segmentación en formato JSON.
M. Obtener segmentación en formato MoB HTML.
N. Obtener segmentación en formato V-PRIMA.

Se presentará la descripción de los servicios y su forma de uso. Las respuestas del servicio consiste en un objeto JSON, todas las respuestas poseen un **código**, si el mismo posee un valor de “200” significa que la petición se realizó de forma exitosa, en caso de ser “400” es porque hubo un error.

A. Registrar usuario: Permite registrar a un usuario en el sistema.

url: https://mob.ciens.ucv.ve/api/users , **método:** POST.

Ejemplo de objeto JSON de la petición:

```
obj = {
'email': 'juan@gmail.com',
'username': 'Juan','
'password': 'miclave123'
}
```

Ejemplo de respuesta:

```
obj = {
'code':200,
'msg':'The user has been registered succesfully'
}
```

B. **Iniciar sesión:** Permite al usuario registrado (y activado) iniciar sesión en el sistema para hacer uso de sus funcionalidades.

url: https://mob.ciens.ucv.ve/api/login , **método:** POST.

Ejemplo de objeto JSON de la petición:

```
obj = {
'username': 'Juan',
'password': 'miclave123'
}
```

Ejemplo de respuesta:

```
obj = {
'code':200,
'msg':'Welcome Juan!',
'username': 'Juan',
'name': 'Juan',
'lastname': 'Perez',
'email': 'juan@gmail.com'
}
```

C. **Cerrar sesión:** Permite al usuario identificado dentro del sistema salir del mismo.

url: https://mob.ciens.ucv.ve/api/logout , **método:** GET.

Ejemplo de respuesta:

```
obj = {
'code':200,
'msg':'you have been logged out'
}
```

D. **Recuperar contraseña:** Permite al usuario recuperar su contraseña en caso de extravío.

   url: https://mob.ciens.ucv.ve/api/recover , **método:** POST.

   Ejemplo de objeto JSON de la petición:

```
obj = {
'email': 'juan@gmail.com'
}
```

   Ejemplo de respuesta:

```
obj = {
'code':200,
'msg':'Your data has been sent to your email.'
}
```

E. **Cargar segmentación:** Este es uno de los servicios más importante del API pues representa la base de todo el sistema, permite cargar los resultados de una segmentación a la base de datos (y los datos de la página Web en caso de que sea la primera vez que se segmenta).

   url: https://mob.ciens.ucv.ve/api/recover , **método:** POST.

Ejemplo de objeto JSON de la petición:

```
obj = {
'collection': 'col_name', (opcional)
'category': 'cat_name', (opcional)
'capture': 'http://www.url_imagen_representativa.jpg', (opcional)
'title': 'page_title', (opcional)
'url': 'http://www.mipagina.com',
'width': 800, (px | opcional)
'height': 1600, (px |opcional)
'gran': 5,
'blocks': BlocksObj,**
'browser': 'nombre_browser',
'seg_type': 'tipo_algoritmo'
}
```

Leyenda:

collection: la colección a la que pertenece la página.
category: la categoría de la página.
capture: la url de la imagen que representa la página.
title: el título de la página.
url: la url de la página.
width: el ancho de la página, en pixeles.
height: el alto de la página, en pixeles.
gran: la granularidad de la segmentación.
blocks: el objeto JSON con los bloques de la segmentación (en el siguiente bloque de código se describe su estructura.)
browser: el navegador usado.
seg_type: el nombre del algoritmo aplicado.

```
** BlocksObj : {
meta : {
'cantblock': 12,
'areablock': 2000
}
block: [
{'id_block': 'id1', 'tag': 'etiqueta1', 'left': 10, 'top': 20,
'width': 200, 'height': 100, 'words': 50, 'elems': 4, ‘gran’: 5,
```

```
‘dom’: DomElemsObj ***},
....
]
}
```

Leyenda:

cantblock: es la cantidad de bloques presentes en la segmentación.
areablock: es el área total de todos los bloques sumadas (en px).
id_block: el id del bloque.
tag: la etiqueta del bloque.
left: la posición left del bloque, en pixeles.
top: la posición top del bloque, en pixeles.
width: el ancho del bloque, en pixeles.
height: altura del bloque, en pixeles.
words: cantidad de palabras dentro del bloque.
elems: cantidad de elementos DOM dentro del bloque.
gran: indica la granularidad del bloque.
dom: un objeto JSON que posee los elementos del dom dentro del bloque (en el siguiente bloque de código se describe su estructura.)

```
*** DomElemObj: {
'meta': {
  'atri1' : 'valor1',
  'atri2' :'valor2',
  ...
},
'content': [
  { 'meta': {}, 'content': ...   }
  ....
]
}
```

Leyenda:

meta: es un JSON con todos los atributos del elemento DOM.
content: es una lista con el mismo formato de dato que DOMElemObj pero aplicado a los hijos del elemento DOM, recursivamente.

Ejemplo de respuesta:

```
obj = {
'code':200,
'msg':'upload completed successfully"
}
```

F. **Obtener vista previa de segmentación:** Devuelve un HTML con un canvas donde se dibujan las figuras y etiquetas de los bloques segmentados para una segmentación en específico. <p_id> indica el id de la segmentación, <zoom> el nivel de zoom que se desee del canvas (4 es buena opción).

url: https://mob.ciens.ucv.ve/api/segpreview/<p_id>/<zoom> , **método:** GET.

G. **Obtener lista de colecciones:** Permite obtener una lista con los nombres de las colecciones y categorías de éstas existentes en la base de datos del sistema.

url: https://mob.ciens.ucv.ve/api/getcollections , **método:** GET.

Ejemplo de respuesta:

```
obj = {
'code':200,
"data": [
[1(id), "Col_name1", "Cat1, Cat2, Cat3"],
[2(id), "Col_name2", "Cat1, Cat2"]
...
]
}
```

H. **Obtener lista de etiquetas:** Permite obtener una lista con los nombres de las etiquetas existentes en la base de datos del sistema.

url: https://mob.ciens.ucv.ve/api/gettags , **método:** GET.

Ejemplo de respuesta:

```
obj = {
'code':200,
"data": [
[1(id), "tag_name1", "desc_esp","desc_eng", "desc_fra" ],
[4(id), "tag_name2", "desc_esp","desc_eng", "desc_fra" ]
...
]
}
```

I. **Obtener lista de puntajes globales:** Permite obtener una lista con los mejores puntajes en cada una de las granularidades de una página Web específica.

url: https://mob.ciens.ucv.ve/api/globalscore , **método:** POST.

Ejemplo de objeto JSON de la petición:

```
obj = {
'url': 'http://www.asamplewebpage.com'
}
```

Ejemplo de respuesta:

```
obj = {
'code':200,
'data': [
{'username': 'Juan' , 'score': 10 },
{'username': '----' , 'score': 0 }
...
]
```

```
}
```

Nota: dependiendo de la posición del json dentro del arreglo, es su granularidad, en el ejemplo, Juan tuvo una puntuación de 10 en su segmentación con granularidad 0. Mientras que nadie ha segmentado la página en su granularidad 1.

J. **Obtener lista de puntajes individuales:** Permite obtener una lista con los puntajes en cada una de las granularidades de una página Web específica para un usuario determinado.

url: https://mob.ciens.ucv.ve/api/userscore , **método:** POST.

Ejemplo de objeto JSON de la petición:

```
obj = {
'url': 'http://www.asamplewebpage.com',
'username': 'Juan'
}
```

Ejemplo de respuesta:

```
obj = {
'code':200,
'data': [0,10,0,2,0,0,0,0,0,0,0]
}
```

Nota: dependiendo de la posición dentro del arreglo, es el puntaje obtenido en una granularidad específica, en el ejemplo, Juan tuvo una puntuación de 10 en su segmentación con granularidad 1, y un puntaje de 2 en la granularidad 3.

K. **Obtener página Web en formato WARC:** Devuelve la información de una página Web en formato WARC (Web ARChive). <id> indica el id de la página Web.

url: https://mob.ciens.ucv.ve/api/pa:<id>.warc.gz , **método:** GET.

L. **Obtener segmentación en formato JSON:** Retorna un JSON con todos los datos de una segmentación en específico. <id> indica el id de la segmentación.

   url: https://mob.ciens.ucv.ve/api/seg:<id>.js , **método:** GET.

M. **Obtener segmentación en formato MoB HTML:** Dada una segmentación determinada, retorna un HTML con la información que se capturó momentos antes de enviar la segmentación, es decir, el HTML original de la página Web modificado por la herramienta MoB tras realizar la segmentación. <id> indica el id de la segmentación.

   url: https://mob.ciens.ucv.ve/api/seg:<id>_mob.html , **método:** GET.

N. **Obtener segmentación en formato V-PRIMA:** Retorna todos los datos de una segmentación específica en formato V-PRIMA, el formato consta de un XML donde se especifican los bloques existentes en la segmentación y los links, imágenes y textos existentes dentro de éstos. <id> indica el id de la segmentación.

   url: https://mob.ciens.ucv.ve/api/seg:<id>_vprima.xml , **método:** GET.

REFERENCIAS BIBLIOGRÁFICAS

Bibliothèque Nationale de France. (2017). The WARC File Format. Consultado en Enero-2018. Recuperado de: https://goo.gl/xf3BEU

Burget, Radek. (2014). FitLayout: Web Page Analysis Framework. Consultado en Abril-2017. Recuperado de: https://goo.gl/JUKYDj

Cabanac G., Chevalier M., Chrisment C., Julien C. (2009). Social Validation of Collective Annotations: Definition and Experiment. Consultado en Junio-2017. Recuperado de: https://goo.gl/LWYf7U

Cai D, Shipeng Y, Ji-Rong W, Wei-Ying M. (2003). VIPS: a Vision-based Page Segmentation Algorithm. Consultado en Marzo-2017. Recuperado de: https://goo.gl/1FNChD

Chakrabarti D., Mital M., Hajela S., Velipasaoglu E. (2008). Automatic Visual Segmentation of Webpages. Consultado en Marzo-2017. Recuperado de: https://goo.gl/Im1z4L

Chrome Developer. (2018). Develop Extensions. Consultado en Septiembre-2017. Recuperado de: https://tinyurl.com/nqfk2za

Egnor, Daniel. (2004). Document Segmentation Based on Visual Gaps. Consultado en Marzo-2017. Recuperado de: https://goo.gl/HS2iUS

Fishkin, Rand. (2010). All Links are not Created Equal: 10 Illustrations on Search Engines' Valuation of Links. Consultado en Marzo-2017. Recuperado de: https://goo.gl/qGrncB

Flask. (2010). Flask User's Guide and Documentation. Consultado en Octubre-2017. Recuperado de: http://flask.pocoo.org/docs/0.12/

González J., Cordero J. (2001). Diseño de Páginas Web. Madrid: McGraw Hill Interamericana Editores, S.A. Consultado en Marzo-2017.

Henry, Shawn Lawton. (2002). Understanding Web Accessibility. En Constructing Accessible Web Sites. ISBN: 1904151000. Consultado en Abril-2017. Recuperado de: https://goo.gl/oT8l1T

Huarachl, Maritza. (2009). Metodo Agil: ASD (Adaptive Software Development). Consultado en Junio-2017. Recuperado de: https://goo.gl/Rt8p1G

International Standard (1991). ISO 9126. Software engineering-Product Quality. Consultado en Abril-2017.

JQuery. (2018). JQuery API. Consultado en Octubre-2017. Recuperado de: http://api.jquery.com/

Kasteler, Jordan. (2011). Page Segmentation and the Effect on Link Building. Consultado en Marzo-2017. Recuperado de: https://goo.gl/lKVcyE

Kearney M., Basques K. (2017). Chrome DevTools. Consultado en Abril-2017. Recuperado de: https://goo.gl/OzaTyv

Kohlschütter C., Nejdl W. (2008). A Densitometric Approach to Web Page Segmentation. Consultado en Marzo-2017. Recuperado de: https://goo.gl/WiC2Fz

Nielsen, Jakob. (2001). Beyond Accessibility: Treating People with Disabilities as People. Consultado en Abril-2017. Recuperado de: https://goo.gl/qrlt2W

Pennock, Maureen (2013). *Web-Archiving*. DPC Technology Watch Reports. Great Britain: Digital Preservation Coalition. doi:10.7207/twr13-01. ISSN 2048-7916. Consultado en Mayo-2017.

Postgis. (2018). PostGIS 2.4.5dev Manual. Consultado en Noviembre-2017. Recuperado de: https://postgis.net/docs/

PostgreSQL. (2018). PostgreSQL 10.4 Documentation. Consultado en Noviembre-2017. Recuperado de: https://tinyurl.com/y7pnxt9a

Python. (2018). Python 3.5 Documentation. Consultado en Octubre-2017. Recuperado de: https://docs.python.org/3.5/

Real Academia Española. (2005) . Diccionario Panhispánico de Dudas. Consultado en Abril-2017. Recuperado de: https://goo.gl/SBr586

Sanoja A., Gançarski S. (2016). Block-o-Matic (Beta) a Web Page Segmentation Algorithm. Consultado en Abril-2017. Recuperado de: https://goo.gl/cbENJm

Sanoja A., Gançarski S. (2016) Block-based Migration from HTML4 Standard to HTML5 Standard in the Context of Web Archives. Proceedings of SCTC'2016. Consultado en Abril-2017. Recuperado de: https://goo.gl/vjIHB8

Sanoja A., Gançarski S. (2015). Web page segmentation evaluation. Proceedings of the 30th Annual ACM Symposium on Applied Computing. Consultado en Abril-2017. Recuperado de: https://goo.gl/RIeALU

Sanoja A., Gançarski S. (2014). Block-o-matic: A web page segmentation framework. Multimedia Computing and Systems (ICMCS). Consultado en Abril-2017. Recuperado de: https://goo.gl/gvcb0q

Sanoja A., Gançarski S. (2013). Block-o-Matic: a Web Page Segmentation Tool and its Evaluation. BDA 2013. Consultado en Abril-2017. Recuperado de: https://goo.gl/ZXisOE

Sanoja A., Gançarski S. (2012). Yet Another Hybrid Segmentation Tool. iPRES 2012. Consultado en Abril-2017. Recuperado de: https://goo.gl/FxqG7T

SCAPE Project. (2014). SCAPE: Scalable Preservation Environments. Consultado en Abril-2017. Recuperado de: https://goo.gl/9BjYmm

Shukla, Amitabh.(2009). What is Page Segmentation?. Consultado en Marzo-2017. Recuperado de: https://goo.gl/ppwW5K

Stéphane Gançarski y Matthieu Cord. (2012). Pagelyzer. Consultado en Abril-2017. Recuperado de: https://goo.gl/DtU1ze

The PHP Group. (2017). PHP: Documentation. Consultado en Abril-2017. Recuperado de: https://goo.gl/IauRXi

Trello. (2017). Trello Help. Consultado en Abril-2017. Recuperado de: https://goo.gl/eCBCj5

Ubuntu-es. (2017). Wget. Consultado en Febrero-2018. Recuperado de: https://goo.gl/AjVdtv

W3C. (2014). W3C Recommendation-HTML5. Consultado en Mayo-2017. Recuperado de: https://goo.gl/84qY8s

W3C. (2005). Document Object Model (DOM). Consultado en Marzo-2017. Recuperado de: https://goo.gl/9Xwtqg

W3C. (2005). Introducción a la Accesibilidad Web. Consultado en Abril-2017 Recuperado de: https://goo.gl/YC1HqE