

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO - CHILE



**“AUTOMATIZACIÓN DE EVALUACIÓN HEURÍSTICA DE
PÁGINAS WEB”**

SVEN HOLLSTEIN

**MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO EN EJECUCIÓN EN INFORMÁTICA**

Profesor Guía: Liubov Dombrovskaia
Profesor Correferente: Roberto León

2023

RESUMEN

Resumen— En esta investigación se buscó automatizar la evaluación heurística de páginas web. Lo anterior se llevó a cabo mediante el desarrollo de una herramienta capaz de entregar resultados correctos e informados sobre el diseño y la organización de una página web en específico. La solución planteada permite al usuario evaluar una página web de manera automática, otorgándole información sobre la página en términos de diseño y usabilidad sin necesariamente solicitar un mayor entendimiento de las técnicas de análisis de usabilidad utilizadas.

Palabras Clave— Usabilidad; Diseño; Automatización; Análisis; Heurísticas.

ABSTRACT

Abstract— This research tried to automate the heuristic evaluation of webpages. The former was accomplished by developing a tool capable of delivering precise and informed results to the user, regarding design and the overall organization of the analysed webpage. The proposed solution allows the user to evaluate a webpage automatically, delivering design and usability information to the user without requesting the user to be highly knowledgeable in usability design analysis.

Keywords— *Usability; Design; Automatization; Analysis; Heuristics.*

GLOSARIO

CSS: *Cascading Style Sheets*. Lenguaje de programación utilizado para dar estilo a un documento HTML.

CTA: *Call To Action*. Término utilizado para describir elementos de una página web que llamen a la acción, como botones o enlaces.

HTML: *HyperText Markup Language*. Lenguaje de programación utilizado en la elaboración de páginas web.

SQL: *Structured Query Language*. Lenguaje de consulta para bases de datos.

UI: *User Interface* (Interfaz de Usuario).

UTFSM: Universidad Técnica Federico Santa María.

UX: *User Experiencie* (Experiencia de Usuario).

INDICE DE CONTENIDOS

RESUMEN	2
ABSTRACT	2
INDICE DE FIGURAS	6
INDICE DE TABLAS	7
INTRODUCCIÓN	8
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	10
1.1 OBJETIVOS Y ALCANCE	11
CAPÍTULO 2: MARCO CONCEPTUAL	13
3.1 ESTADO DEL ARTE.....	22
CAPÍTULO 3: PROPUESTA DE SOLUCION	26
3.1 IMPLEMENTACIÓN	26
3.2 BASE DE DATOS	27
3.2.1 MODELO BD.....	27
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN	33
4.1 CARACTERÍSTICAS AUTOMATIZADAS	33
4.1.1 ASPECTOS GENERALES.....	33
4.1.2 HEADER, FOOTER, SEARCH Y LOGO.....	34
4.1.3 LETRA Y COLOR.....	35
4.1.4 ENLACES.....	36
4.1.5 PUBLICIDAD Y MEDIA	37
4.2 RESULTADOS.....	38
4.3 CONVENCIÓN DE DISEÑO DE PÁGINAS WEB CHILENAS	40
CAPÍTULO 5: CONCLUSIONES	44
5.1 RECOMENDACIONES	47
REFERENCIAS BIBLIOGRÁFICAS.....	49
ANEXOS	51

1. CONVENCIONES VIGENTES.....	51
2 IMPLEMENTACIÓN	59
2.1 Analizador de páginas web.....	59
2.2 Llenado de datos para confección de parametría	74
2.3 Creación de parámetros	80
3 RESULTADO GRÁFICO EVALUACIÓN	86
4 RESULTADO GRÁFICO PARÁMETROS.....	87

INDICE DE FIGURAS

Figura 1: Modelo Base de Datos..... 27

INDICE DE TABLAS

Tabla 1: Aspectos Generales	33
Tabla 2: Header, Footer, Search y Logo.....	34
Tabla 3: Letra y Color	35
Tabla 4: Enlaces	36
Tabla 5: Publicidad y Media.....	37
Tabla 6: Resultados obtenidos para https://www.paris.cl/	38
Tabla 7: Resultados obtenidos para el análisis de la muestra (objetivo secundario).....	40

INTRODUCCIÓN

El internet ha otorgado al mundo un sinnúmero de beneficios respecto a maneras de presentar información en formatos digitales, siendo el más común una página web. Lo anterior ha significado para muchos una mejora en términos de alcance y accesibilidad, permitiendo a todo tipo de entidades y personas entregar información a un público que no habría podido acceder a ella de no existir esta tecnología.

Si bien los cambios que nos ha traído el avance tecnológico suelen ser en su mayoría positivos, vemos en el mundo de las páginas web una serie de elementos que, con el paso del tiempo, han sido evaluados y criticados tanto por simples usuarios como por expertos en diseño.

El problema planteado en esta memoria es precisamente el cómo determinar qué tan efectiva puede llegar a ser una página web en su objetivo de llegar al público deseado, tomando en cuenta distintos factores y basándose en convenciones internacionales de diseño, acotadas al actual mercado de páginas web chilenas.

Se identifica lo anterior como un problema, dado que se observa que, si bien el usuario promedio entiende cómo utilizar una página web, no tiene necesariamente los conocimientos para determinar qué tan efectiva es dicha página en entregar información o realizar una acción determinada.

Como base conceptual para la explicación del problema y la búsqueda de una solución se tiene el trabajo del Dr. Jakob Nielsen, cuyos escritos serán citados en variadas oportunidades durante el desarrollo de la investigación.

La evaluación heurística es un proceso que consiste en mirar una interfaz e intentar formarse una opinión sobre lo bueno y lo malo de ella. Dice que es probable que la mayoría de las personas realicen algún tipo de evaluación heurística usando su propia intuición y sentido común (Nielsen, 1993).

También, plantea que el objetivo de la evaluación heurística es encontrar problemas de usabilidad en un diseño de interfaz de usuario, para que sean incluidos dentro del proceso iterativo de diseño.

De aquí en más se hablará de las heurísticas para referirse a los conceptos desarrollados por el autor en sus varios escritos y publicaciones, que servirán de base para la investigación llevada a cabo.

El tener en cuenta las heurísticas de Nielsen y las distintas convenciones que existen a nivel internacional al momento de desarrollar una página web, puede resultar ser una herramienta muy útil para un desarrollador. Le permiten tener en cuenta que características de la página debe desarrollar de qué manera, para lograr que estas cumplan su objetivo de la mejor manera posible.

Además de los posibles desarrolladores, el saber de heurísticas también puede servir a simples usuarios de estas plataformas, dado que les permiten tener un ojo más crítico respecto a los medios que están utilizando y así poder identificar páginas bien configuradas.

Para las personas que están interesadas en que su empresa o negocio tenga una página exitosa que les ayude a concretar ventas o les sirva como un medio de contacto efectivo con sus clientes, puede resultar de igual manera atractivo el interiorizarse en estos temas. Lo anterior posibilita un mejor entendimiento de las posibilidades existentes al momento de desarrollar y saber que solicitar al desarrollador.

La solución que será explicada en este escrito intentará saber hasta qué punto se puede automatizar la evaluación heurística de páginas web.

Lo anterior se llevará a cabo mediante el desarrollo de una herramienta que permitirá al usuario ingresar la URL de la página que desea evaluar, lo que le entregará un informe detallando los elementos encontrados en la página, con resultados numéricos y comentarios para contextualizar.

De esta manera el usuario dispondrá de una lista de elementos y de qué manera son percibidos estos elementos por la herramienta, permitiéndole entender en que puede estar fallando la página evaluada y que está llevando a cabo de manera correcta.

Para evaluar la página web se dispondrá de una lista de parámetros. En primer lugar, se utilizará un conjunto de convenciones según lo estudiado en (Toro, 2015), que fue aplicado a las 50 páginas más visitadas en Chile y generó un set de convenciones. De estas convenciones se extraerá parámetros concretos para ser utilizados en el análisis de la herramienta.

Junto con lo anterior se dispondrá de un analizador de páginas web que creará un set de parámetros basándose en diversas pruebas que le serán realizadas a una muestra compuesta por las 50 páginas web chilenas más visitadas. Estos parámetros serán utilizados para ser puntos de comparación en el análisis de otras páginas.

El documento se estructura como sigue. En primer lugar, se definirá el problema. En el marco conceptual se presentará las convenciones mencionadas anteriormente, así como los distintos autores y temas que ayudarán a contextualizar el problema y la solución.

Los capítulos siguientes explicarán al lector como fue llevada a cabo la solución y el nivel de efectividad que se alcanzó respecto a los objetivos planteados. Se terminará el cuerpo del escrito con las conclusiones obtenidas luego del desarrollo de la solución, así como posibles recomendaciones para el lector, en caso de que desee continuar la línea investigativa planteada.

CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA

El diseño de páginas web y nuestro entendimiento de este ha evolucionado y avanzado con el paso del tiempo, desde la primera página web hace ya más de 30 años a las múltiples páginas nuevas que salen al mundo cada día.

Esta evolución ha traído consigo cambios significativos, tanto en diseño como en funcionamiento y público objetivo. Empresas y particulares han identificado con el tiempo que formas de diseño cumplirán con el objetivo que desean para su página, sea una página informativa, un foro o un *Marketplace*.

Se ha desarrollado una serie de ideas de diseño que han evolucionado desde simples decisiones personales del desarrollador a convenciones internacionales, las que toman en cuenta muchos factores para definir la mejor manera de organizar el contenido de una página web. Lo anterior se puede ver reflejado en múltiples escritos y publicaciones de diseñadores de todo el mundo, lo que será explorado en el capítulo siguiente.

Una de las nuevas ideas que está marcando la pauta hoy en día es la relación UX – UI. UX significa “*User eXperience*” o experiencia de usuario y UI significa “*User Interface*” o interfaz de usuario, ambos vistos desde el punto de vista del diseño.

Según (Lamprecht, 2022) ambos son cruciales y trabajan juntos, ya que la experiencia de usuario abarca todos los aspectos de la interacción del usuario final con la compañía, sus servicios y sus productos.

Respecto al UI, plantea que se debe entender como el complemento del UX. Cuando el UX es un conglomerado de tareas enfocadas en la optimización y el rendimiento de un producto para un uso agradable, el diseño de interfaces de usuario tiene que ver con la presentación de un producto, con los aspectos interactivos, con el “*look and feel*” de una página web.

Si bien se entiende el UX como el aspecto fundamental dado que la página debe funcionar, el UI es igual de importante dado que si la página no es atractiva, no llegará al público que necesita. Dado que el UI tiene un aspecto fuerte de diseño y decisiones visuales, es que debe ser evaluado para dar con un producto satisfactorio en términos funcionales, pero a su vez atractivo en términos visuales.

Actualmente existen guías y listas sobre convenciones de diseño, las que explican y ejemplifican el cómo se debería llevar a cabo el desarrollo, dependiendo de lo que se quiere lograr con la página. A modo de ejemplo de herramientas que existen hoy en día para revisar código de páginas web, se tienen herramientas de W3C de Validación de CSS (W3C, 2019) y HTML (W3C, 2019).

Otra de las maneras de evaluar el UI es a través de la evaluación heurística, donde podemos tomar como ejemplo las 10 heurísticas que plantea Nielsen en (Nielsen, 2002), las cuales están pensadas para ser usadas para evaluar una página en términos cualitativos.

Otro ejemplo que usa este alcance para evaluar páginas es el planteado por David Travis en (Travis, 2016). Esta guía permite evaluar cuantitativamente el diseño de una página web, mediante una gran cantidad de puntos que el usuario debe evaluar mientras navega por el sitio en cuestión.

Las reglas presentadas por los autores mencionados representan convenciones de diseño que han sido probadas prácticamente en el diseño de páginas web. La elección del set de reglas que el evaluador decida utilizar dependerá del tipo de sitio y de la experiencia del evaluador mismo.

Dado lo anterior es que se plantea la problemática a solucionar en esta investigación.

Específicamente, se desea investigar la posibilidad de crear una herramienta capaz de realizar una evaluación cuantitativa y cualitativa de páginas web en tiempo real y sin la necesidad de conocimientos de ningún tipo de parte del usuario.

En concreto, se buscará responder a la pregunta: ¿Hasta qué punto se puede automatizar la evaluación heurística?

Se identifica el problema como el método que se tiene actualmente para evaluar páginas web y como este podría ser mejorado, tomando en cuenta un sistema que automatice esta evaluación y entregue al usuario un resultado que le permita entender lo que necesita entender.

La herramienta planteada en este escrito pretenderá ser atractiva tanto para desarrolladores como para personas naturales, dado que no requerirá de conocimiento técnico para su uso y entregará información precisa y concisa sobre el diseño de la página en cuestión. De no llevarse a cabo esta investigación y el posterior desarrollo de la herramienta, seguirán estando los lineamientos y las convenciones a disposición de los desarrolladores, pero tendrán que realizar este análisis ellos mismos.

Por otro lado, la herramienta permitirá mejorar muchas páginas que ya se encuentran funcionales y que, por una o varias decisiones erróneas en términos de diseño, no cumplen su objetivo o carecen de muchas visitas.

A continuación, se precisa los objetivos y alcances de la memoria.

1.1 OBJETIVOS Y ALCANCE

Se identifican dos tipos de objetivos a ser cumplidos en esta investigación, un objetivo principal y un objetivo secundario.

El objetivo principal es el desarrollo de una herramienta de un uso simple e intuitivo, que entregue resultados sobre el aspecto general de una página web en términos de diseño, contenido y aspecto general, para informar al usuario sobre posibles modificaciones que sería recomendable llevar a cabo en la página evaluada para mejorar el número de visitas y de consultas satisfactorias.

El objetivo secundario es crear un set de convenciones para páginas web chilenas, sacado del análisis de las 50 páginas más visitadas en Chile durante el período de desarrollo.

Esta investigación buscará desarrollar una herramienta robusta y simple, que permita al usuario evaluar la página de inicio de páginas web chilenas. Lo anterior se logrará mediante un software que solicite una URL al usuario y entregue resultados luego de una breve espera. Los resultados entregados deberán ser lo suficientemente simples y certeros, de manera que la herramienta sea accesible para todo tipo de personas.

Se intentará abarcar una cantidad razonable de elementos de diseño, lo que posibilitará la entrega de resultados efectivos.

CAPÍTULO 2: MARCO CONCEPTUAL

En el libro (Nielsen, 1993) se define al proceso de evaluación heurística como: “la Evaluación Heurística se lleva a cabo mirando una interfaz e intentando formarse una opinión sobre lo bueno y lo malo de ella”. Con esta descripción Nielsen explica que este proceso es realizado por un observador con una mirada subjetiva, que intentará evaluar lo observado bajo parámetros que tal vez el mismo definió.

Es probable que la mayoría de las personas realicen algún tipo de evaluación heurística usando su propia intuición y sentido común. Idea que viene a respaldar la idea planteada anteriormente de una mirada subjetiva.

El objetivo de la evaluación heurística es encontrar problemas de usabilidad en un diseño de interfaz de usuario, para que sean incluidos dentro del proceso iterativo de diseño. Con esto define que el proceso de evaluación heurística puede ser llevado a cabo las veces que sea necesario, con tal de encontrar todos los posibles problemas que pueda presentar la interfaz.

Luego de dejar esto claro, explica el funcionamiento de la evaluación de la siguiente forma: la evaluación heurística comprende tener a un pequeño grupo de evaluadores examinar la interfaz y juzgar su cumplimiento con principios de usabilidad conocidos (las “heurísticas”). Luego entrega un set típico de heurísticas presentadas como capítulos en el libro:

- Dialogo simple y natural: lo ideal es presentar exactamente la información que el usuario necesita, justo cuando lo necesita y en el lugar adecuado.
- Hablar el idioma del usuario, tomando en cuenta el público objetivo y sus capacidades. Se debe intentar usar el idioma y la terminología correctos, para evitar confundir.
- Minimizar la carga de memoria para el usuario: es más fácil dejar que el usuario modifique información presentada por el sistema, que obligar al usuario a completar formularios recordando todo el mismo.
- Consistencia: la misma información debería presentarse en la misma ubicación en todas las pantallas y cuadros de diálogo y se debería organizar de la misma manera para facilitar el reconocimiento por parte del usuario.
- Retroalimentación: el sistema debería informar continuamente al usuario sobre lo que está haciendo y como está interpretando el ingreso de información. Como ejemplo plantea el corregir en tiempo real cuando una palabra es escrita erróneamente.

- Proveer salidas: el sistema debería presentar continuamente una opción al usuario de terminar lo que está haciendo y dejarlo salir o volver, para otorgarle un sentimiento de control sobre la herramienta y no hacerle sentir atrapado.
- Accesos directos: si bien el sistema debería ser simple de usar con unas cuantas reglas de conocimiento general, es bueno que contenga atajos para usuarios experimentados que le permitan ahorrarse algunos pasos.
- Buenos mensajes de error: Los mensajes de error deberían estar formulados en un lenguaje claro, ser precisos en vez de vagos o generales, deberían ayudar al usuario a resolver el problema y no contener lenguaje abusivo.

Estas heurísticas son reglas generales que parecen describir propiedades comunes de interfaces usables. Son reglas que han sido acuñadas luego de observar e investigar la interacción de múltiples usuarios con los distintos tipos de sistemas que se tiene actualmente.

El resultado de aplicar estas reglas con el método de evaluación heurística es una lista de problemas de usabilidad en la interfaz, anotados con referencias a los principios de usabilidad que fueron violados por el diseño en cada caso, según la opinión de un evaluador.

Se entiende, entonces, el proceso de evaluación heurística de Nielsen como un proceso largo y complicado, que debe ser llevado a cabo por un encargado que entienda como se ejecuta junto a evaluadores que serán los responsables de realizar juicios de valor respecto a lo que observan, tomando en cuenta un set de reglas preestablecidas que les servirán para dirigir la mirada a los elementos que el evaluador desea investigar. Esto se realiza entonces, con el objetivo de encontrar problemas en la usabilidad y el diseño de una interfaz para que sean resueltos durante el ciclo de desarrollo del sistema.

Complementando lo expuesto por Nielsen en el libro anteriormente citado, se tiene otro escrito del autor en (Nielsen, 2002). En este artículo explora las 10 convenciones más importantes, que se debe tener en consideración para incrementar la usabilidad de la página principal. Estas convenciones se listan y explican a continuación:

1. Comenzar el contenido de la página con un eslogan que resuma lo que el sitio hace, con la idea de atraer nuevos clientes.
2. Comenzar el contenido del Título (TITLE Tag) con el nombre de la compañía, acompañado de una breve descripción. Lo anterior para situar bien la página en herramientas de búsqueda.
3. Se debe dar a conocer toda la información que el cliente pueda desear saber sobre la compañía. Lo recomendable es dejarlo en una sección aparte y no llenar la página principal con esta información.

4. La página principal debería entregar direcciones al usuario sobre las tareas principales que puede llevar a cabo en el sitio.
5. Se debe incluir un cuadro de búsqueda, el que se recomienda sea por lo menos de 27 caracteres de ancho.
6. Mostrar contenido del sitio en la página principal, para incitar a su exploración.
7. Los links sirven como puertas hacia el contenido a encontrar en el sitio. Es importante empezar cada link con una palabra importante que ayude al usuario a encontrarlo y entender para que sirve con un escaneo general de la página y no tener que obligarlo a buscar con detención.
8. Los usuarios recordarán productos, artículos o promociones que hayan sido promovidos en la página principal, pero no lo sabrán encontrar cuando dejen de ser promocionados. Se recomienda mantener una lista de las últimas promociones en la página principal, aparte de un archivo donde se puedan encontrar todas las promociones antiguas.
9. Los usuarios suelen pasar de largo de gráficos porque creen que son anuncios. Se recomienda mantener los ítems importantes de la página principal simples y no llenos de ilustraciones o colores.
10. Se recomienda usar ilustraciones reales que estén conectadas con lo que se ofrece y evitar utilizar imágenes que no presten más información al usuario.

La página principal es la página más importante en la mayoría de los sitios web y recibe más visitas que cualquier otra.

Se utilizará esta lista de heurísticas para considerar las posibles reglas a ser automatizadas en el desarrollo de la herramienta propuesta. De las mencionadas se puede notar a simple vista que la posibilidad de automatizarlas no resultaría fácil, sin mencionar las que no se podrían automatizar dado que dependen de una vista humana subjetiva o dependen del análisis de texto que se escapa de las posibles funcionalidades de esta herramienta.

De las anteriores se puede desprender, por ejemplo, la necesidad de incluir un cuadro de búsqueda. Esta heurística es posible de automatizar directamente dado que es buscar un elemento en la página web. Al contrario, se puede mencionar que el agrupar toda la información importante respecto a la empresa en un solo lugar no es posible de automatizar o se escapa del alcance de la herramienta propuesta, dado que implica saber con anterioridad a quien pertenece la página web a analizar, además de la necesidad de leer e interpretar texto.

En el estudio (Toro, 2015), se analizaron las páginas de inicio de las 50 páginas chilenas más visitadas. Se decide utilizar la misma muestra en esta investigación, dado que permitirá obtener resultados reales y útiles respecto al actual mercado de páginas web chilenas.

Este estudio de usabilidad se realizó con un cuestionario para recolectar datos sobre las características de los 50 sitios más visitados en Chile, analizar los datos obtenidos y consolidar sus resultados en un set de convenciones.

A continuación, se presentan las características evaluadas en cada página y los aspectos que se consideraron de cada uno:

1. Tiempo de Descarga: Menciona que el tiempo medio de descarga para las páginas de la muestra fue de 5 segundos, lo que está dentro del tiempo máximo de 10 segundos.
2. Ancho de la Página: Explica que la mayoría de las páginas fueron diseñadas para resoluciones de 1280x800 o 1366x768.
3. Longitud de la Página: Menciona que la mayoría de las páginas usa un largo de 2 a 8 longitudes de página en su página de inicio, dado que usan la técnica de infinite scrolling. Debería ser de 1 a 2 largos de página.
4. Diseño Líquido, Estático y Responsivo: La mayoría de las páginas utiliza contenido estático, mientras que cerca de un tercio usa correctamente contenido responsivo. Se debe usar diseño responsivo aun cuando muy pocas páginas lo utilicen efectivamente.
5. Frame vs iFrame: Menciona que todas las páginas han pasado a utilizar iFrame en vez de Frame, tecnología que ha quedado obsoleta.
6. Logo: Todas las páginas contaban con un logo visible. La mayoría lo ubicaba en la esquina superior izquierda, lo que se condice con lo expuesto por Nielsen.
7. Cuadro de Búsqueda: El 24% de las páginas consultadas no tenía cuadro de búsqueda. Dado que los resultados fueron variados sobre su posicionamiento, sigue la recomendación de Nielsen, de ubicarlo en la parte superior de la página. Por otra parte, menciona el tamaño del cuadro, explicando que siendo la mediana de las páginas web de 32 caracteres no se cumple con la recomendación internacional de 35 a 40 caracteres.
8. Navegación: Menciona al menú horizontal como el predominante, junto con recomendar la navegación al pie de página.
9. Mapa del Sitio: Se recomienda incluir un mapa del sitio, solo si ofrece una vista general y mejora la navegación.

10. Página de Enrutamiento: Menciona que es una práctica poco utilizada por páginas comunes, con un 8% de la muestra utilizándolo.
11. Páginas de Bienvenida: Ninguna de las páginas consultadas cuenta con página de bienvenida, por lo que no lo recomienda.
12. Características Frecuentes: Menciona ejemplos como información de la empresa o formularios de contacto.
13. Registrarse: Menciona que la mitad de las páginas ofrece un registro, pero ninguna imposibilita la navegación sin iniciar sesión, lo que es recomendable.
14. Sobre Nosotros: Recomienda la existencia de alguna referencia a datos de la empresa, como un enlace a “Quiénes Somos” o similar.
15. Datos de Contacto: Menciona que todas las páginas cuentan con datos de contacto y recomienda que contengan un enlace a un formulario para contactar directamente.
16. Política de Privacidad: La mayoría de las páginas cuentan con alguna sección que cubra contenidos de este estilo, recomienda llamarla “Política(s) de Privacidad” dado que el 54% de las páginas lo realiza.
17. Ofertas de Empleo: Recomienda la utilización de un “Trabaja con Nosotros” dado que es lo que se encuentra en la mayoría de las páginas.
18. Ayuda: No se recomienda su inclusión en la página de inicio.
19. Imágenes: Menciona que la mediana del número de ilustraciones de la muestra es de 62 distribuidas en más del 15% del tamaño total de la página de inicio, lo que no es recomendable. Sigue la recomendación de Nielsen no saturar la página de inicio para evitar demoras en tiempos de carga y apariencia general.
20. Texto ALT: Recomienda el uso de textos ALT en páginas con gráficos, para su mejor entendimiento por personas con discapacidad visual.
21. Música: Menciona que ningún sitio reproducía música automáticamente, por lo que es lo que recomienda.
22. Animación: Dice que en general no se recomienda la animación, pero que la mayoría de las páginas incluían al menos un elemento animado como por ejemplo un carrusel de imágenes.

23. Publicidad: La mediana del número de anuncios en las páginas que los contenían fue de 7. Nielsen recomienda un máximo de 3.
24. Pop-ups: No se recomienda su uso, pero detectó que la mayoría contaba con estos elementos.
25. Tipografía: Menciona que la mayoría utiliza el color negro para títulos y textos y el blanco para el color de fondo. Además, el 70% de las páginas utilizaban un tamaño de letra de 12 puntos y una fuente de la familia Sans-serif, por lo que esto es lo que recomienda, lo que a su vez sigue lineamientos respecto a lo expuesto por Nielsen.
26. Links: Menciona que el uso recomendado de links es de color azul y subrayados, pero que un alarmante 18% de las páginas cuenta con esa configuración.

De este gran número de características, se tiene algunas que a primera vista resulta posible pensar en su automatización, como lo son la existencia de un Logo o un Cuadro de Búsqueda, el uso correcto de Tipografía o la existencia de Publicidad o Pop-ups. Características como secciones de políticas de privacidad o datos de contacto no resultarían simples de automatizar dado que incluyen la lectura de texto.

Se debe mencionar en esta instancia, que hasta el momento todos los estudios de usabilidad mencionados toman en cuenta un aspecto fundamental: el estudio se realiza a través de la mirada subjetiva de un usuario, que es capaz de identificar y diferenciar los distintos elementos de una página web. Dado que esta investigación busca la automatización de este proceso, es que es necesario acotar el número de funcionalidades que se buscará evaluar.

Lo anterior incide directamente en que pruebas a los aspectos de una página que tengan que ver con el UX no será fácil de automatizar, dado que dependen de la mirada crítica de un usuario y no son características que presenten fácil disponibilidad de ser evaluadas por una máquina.

De las características evaluadas en (Toro, 2015), las siguientes se consideran susceptibles a ser automatizadas con la herramienta propuesta: tiempo de descarga, ancho de la página, longitud de la página, logo, cuadro de búsqueda, música, publicidad, pop-ups y tipografía. Aspectos como el diseño responsivo, la navegación o el mapa del sitio no resultan, luego de una mirada inicial, fáciles de automatizar.

Para dar mayor completitud a lo citado anteriormente, se ofrece otra obra de Nielsen. En este libro (Nielsen, 2001) el autor desarrolla un estudio de usabilidad en 50 páginas principales de páginas obtenidas de listas como las de Fortune 500 o las páginas de mayor tráfico a nivel internacional, que le permite identificar características comunes, las cuáles explica como:

- Utilizar un guion (-) para unir frases importantes, para facilitar su identificación.
- Evitar el uso de signos de exclamación.

- Utilizar el color azul para enlaces no visitados y el color morado para enlaces visitados.
- Si un enlace hace algo más que dirigir a otra página, indicar explícitamente su función.
- No dejar navegación horizontal sobre el banner principal.
- No incluir un enlace a la página principal en la página principal.
- Todas las páginas aparte de la página principal deberían permitir presionar el logo para volver a la página principal.
- No usar espacio de la página principal para dar la bienvenida a usuarios, en vez de esto usar un *tag-line* o slogan.
- Puede ser positivo decorar el sitio con diseño festivo, pero se debe evitar agregar tempranamente.
- En vez de usar números para las fechas, prefiera escribir el mes o su abreviación. (*Ene 2, 2003 vs. 01/02/03*)
- Incluir un cuadro de búsqueda con fondo blanco, de 30 caracteres.

Si se desea identificar rápidamente cuáles de los aspectos nombrados se podría automatizar, es posible señalar que los colores de los enlaces y el cuadro de búsqueda son candidatos inmediatos. Automatizar características como el decorar la página con elementos festivos o la función de los enlaces presenta una complicación mayor, por lo que se pueden descartar en primera instancia.

La autora de (Crestodina, Accedido: 2020) basada en los 50 sitios web internacionales más visitados plantea que existe un set de convenciones web estándar, que corresponde a estándares de diseño y buenas prácticas. Se refiere a un set de reglas que siguen los diseñadores de páginas web, sabiendo que se alinean con las expectativas de los usuarios. Son pautas de claridad y usabilidad.

La diferencia entre estándar, convención y confusión es la siguiente:

- Estándar: El 80% o más de los sitios web usa el mismo tipo de diseño.
- Convención: Entre el 50 y el 79% de los sitios web usan ese diseño.
- Confusión: 49% o menos de los sitios cumple, no domina un tipo de diseño.

Al evaluar los 50 sitios web de marketing con más visitas, obtuvo porcentajes aplicables a la categorización anterior:

1. Logo en la esquina superior izquierda: El 100% de las páginas visitadas contaban con un logo en esta parte de la página, en cada página. Esto corresponde a un Estándar.
2. Contacto en la esquina superior derecha: Solo el 44% de las páginas consultadas tenía un botón o *link* de contacto en esta sección de la página. A pesar de que es considerado como una buena práctica, este estudio no lo cataloga como estándar.
3. Navegación principal en parte superior: El 88% de las páginas visitadas tenían la navegación principal en el "*header*" en la parte superior de todas sus páginas. Por lo

anterior, se considera un estándar el tener navegación en la parte superior de la página.

4. Página principal con presentación: El 32% de las páginas visitadas tenía algún estilo de muestreo cíclico de imágenes o mensajes. Se refiere a que esta práctica no está bien definida entre diseñadores, variando también los resultados obtenidos. Recomienda elegir lo que convenga según el tipo de sitio y los potenciales usuarios y el mensaje que se desee conferir.
5. Propuesta de valor ubicada “arriba” en la página principal: El 80% de los sitios de marketing tiene algún tipo de propuesta de valor explícita en la parte superior de la página principal. Entonces, la mayoría explica su valor al usuario “de entrada”. Las páginas restantes no tenían siquiera una propuesta de valor clara.

Entiéndase como parte superior de la página la parte visible sin tener la necesidad de navegar en la página, es decir, lo que el usuario ve directamente al abrirla.

6. Llamado a acción (CTA – *Call to Action*) “arriba” en la página principal: El 78% de las páginas usadas contenía algún tipo de CTA prominente visualmente en el *home*. No es lo suficiente para ser llamado estándar, pero es definitivamente una convención.
7. Herramienta de búsqueda en el header: el 54% de los sitios visitados contienen una herramienta de búsqueda en la parte superior de la página. Cerca de la mitad de todos los sitios de marketing no poseen una herramienta de búsqueda que aparezca globalmente en la página.

No lo plantea como una sorpresa, dado que una herramienta de búsqueda es necesaria solo cuando el sitio cuente con mucha información. Dice que una herramienta de búsqueda es a veces un intento de organizar una página mal organizada.

8. Casilla de suscripción en el footer: Solo el 24% de las páginas visitadas contienen una forma de suscripción para obtener noticias por mail. El contenido más usual para pies de página es información legal, links de contacto o acuerdos de privacidad. Los usuarios esperan encontrar información de contacto en el pie de página, ya sea en la mitad o en la derecha.
9. Iconos de redes sociales en el pie de página: El 72% de las páginas contienen iconos con links a redes sociales en el footer, haciendo de esta práctica casi un estándar. En contraste, el 26% de las páginas contenían estos iconos en el header. Se recomienda mantener estos iconos en el footer y de un color poco llamativo que solo se ilumine al pasar el ratón por encima, para evitar que llame mucho la atención y pueda provocar que usuarios salgan del sitio anticipadamente.

10. Diseño sensible: El 68% de las páginas son sensibles al tipo de dispositivo con el que se está ingresando a ellas, permitiendo que se vea de manera óptima sea desde un computador o de un celular o Tablet. Es una característica muy difícil de implementar en un sitio ya operativo, lo que puede estar provocando que esta práctica sea solo una convención y no un estándar.

Últimamente el diseño “responsive” se ha vuelto costumbre, dado que la mayoría del tráfico en páginas web es realizado a través de dispositivos móviles hoy en día. Esto último condiciona el actual diseño de páginas web, dado que deben tomar en cuenta un diseño sensible para considerarse competitivos en el mercado y tener alguna opción de éxito.

Aparte del posicionamiento del logo, la navegación principal y la proposición de valor, no hay estándares en diseño web. (Crestodina, Accedido: 2020)

Dentro de las convenciones encontradas se incluyen un prominente llamado a la acción (CTA), herramienta de búsqueda en el header, íconos de redes sociales en el footer y un diseño responsive.

Puede que otras decisiones populares de diseño puedan ser consideradas dentro de las mejores prácticas, pero esto no implica que sean usadas por la mayoría de los sitios web. Un diseño personalizado y específico para la audiencia y el negocio es lo que manda.

Tomando en consideración las obras citadas y la información recopilada sobre heurísticas y convenciones según distintos autores, se intentará crear una lista de convenciones generales a automatizar.

Siendo el objetivo de la herramienta propuesta el disponer de un sistema que efectúe un análisis de usabilidad de manera automática, es que se debe en esta instancia definir que tipos de características se considera posible automatizar para ser buscadas y evaluadas por la herramienta.

Dentro de los aspectos generales que toda página debería tener, se considera importante la existencia de un logo y de un cuadro de búsqueda. Para ambos se intentará identificar su existencia, su tamaño y su funcionalidad. A estos aspectos se podría unir el detectar la existencia de un header y un footer claramente especificados en el código de la página, dado que se considera como buena práctica. Tanto para el header como para el footer se buscará su correcta existencia y su tamaño en relación con el resto de la página.

Continuando con aspectos de diseño susceptibles a ser automatizados, se decide incluir el tamaño de la página, midiendo ancho y alto comparando con el tamaño esperado, además de evaluar el largo de la página, buscando cuantos “largos de página” mide la página principal.

Respecto a parámetros generales de un sitio web, se cree posible automatizar características como el medir el tiempo de carga de la página y buscar videos, audio o anuncios que se reproduzcan automáticamente.

Otro aspecto importante de ser considerado y que presenta buenas posibilidades de ser automatizado es el correcto funcionamiento de enlaces en la página, considerando el diseño y la función que deben cumplir.

Por último, también se cree posible automatizar el análisis de distintos aspectos de la tipografía del sitio, como lo son la fuente del texto utilizada, su tamaño y del color utilizado, tanto de la letra como del fondo.

A modo de información complementaria, se presenta un estudio de convenciones vigentes llevado a cabo al momento de realizar la investigación previa para el desarrollo de esta memoria, lo que se puede encontrar en el Anexo 1.

3.1 ESTADO DEL ARTE

Como fue mencionado anteriormente en este escrito, el ejemplo más directo de comparación son las ideas expuestas por Nielsen en sus escritos, en los cuales da directrices de cómo realizar una evaluación heurística de páginas web. Esta manera específica de realizar un estudio heurístico comprende una serie de necesidades.

Para poder llevar a cabo una evaluación heurística, es necesario contar con un responsable que tenga experiencia y maneje los conceptos necesarios. Lo anterior comprende entender a cabalidad las heurísticas y saber utilizar los conceptos de manera concreta para poder orientar un estudio pueda entregar datos relevantes. Otro aspecto que considera este tipo de evaluación es que necesita de usuarios de prueba y de que estos sean capaces de analizar la página seleccionada siguiendo los criterios preestablecidos por la persona responsable del estudio.

Todo lo anterior hace que el proceso planteado resulte engorroso y que pueda tomar una cantidad de tiempo relevante. Teniendo esto en cuenta es que se identifica la necesidad de intentar automatizar este proceso para así lograr hacerlo más accesible. Dado que la solución planteada en este escrito librará al usuario de necesitar el conocimiento necesario para realizar la evaluación el mismo, es que se considera como una salida válida y útil.

Realizando la investigación sobre lo planteado en este escrito fue posible encontrar un gran número de guías y artículos que se refieren a las heurísticas de Nielsen y como deben ser entendidas para poder ser aplicadas en una evaluación. Las páginas visitadas presentan la información al usuario y permiten comprender el objetivo detrás de este tipo de evaluación, sin embargo, no presentan herramientas automáticas que realicen la evaluación y entreguen información posteriormente al usuario.

Complementando el punto anterior, se encuentra que gran parte de, sino todos los artículos visitados para buscar más contexto sobre la evaluación heurística presentan una visión de cómo deben ser entendidos y aplicados. Un desarrollador que se maneje con conceptos de UX / UI y que entienda la manera de aplicar estos conceptos a un desarrollo podrá también entender y aplicar los conceptos relacionados a la evaluación heurística.

Durante la investigación se encuentra una herramienta llamada Heurix, que pretende posibilitar la realización de una evaluación heurística de manera fácil y rápida, entregando un informe detallado al finalizar la evaluación. (Heurix, 2021).

La herramienta de Heurix presenta una página web que luego de solicitar al usuario crear una cuenta, permite la evaluación de una página web. La manera de realizar esta evaluación es manual, contestando una serie de preguntas sobre distintos tópicos, dentro de los cuáles se encuentran temas como primeras impresiones, navegación, confianza y persuasión e interacción. Luego de contestar un set de 48 preguntas que siguen los lineamientos de las heurísticas de Nielsen, el sistema entrega un informe. Este informe no presenta mucha ayuda al usuario más allá de explicar lo que puede resultar lógico: si en la encuesta se selecciona que no hay cuadro de búsqueda, la herramienta responde con: “es necesario tener un cuadro de búsqueda”.

Luego de probar la herramienta se puede identificar que puede servir para un gran número de personas, que probablemente no tienen todo el conocimiento necesario para desarrollar una página web y necesitan ayuda. Las preguntas que plantea la herramienta se pueden contestar con un sí, no, se puede mejorar y no aplica. Esta manera de contestar las preguntas que plantean presenta inmediatamente la información que se necesita de la herramienta. A modo de ejemplo, si en la encuesta el usuario contesta que el sitio no posee *breadcrumbs* puede identificar inmediatamente que su sitio debería tener esa funcionalidad implementada.

Si bien la herramienta puede resultar útil, no se considera que funcione como una automatización, dado que más allá de requerir la creación de una cuenta, necesita bastante input del usuario y no entrega mucha información al finalizar. Si se compara esta herramienta con la solución planteada se identifica que la solución resultará ser más útil y para un público más grande, dado que no necesita que el usuario comprenda conceptos de evaluación heurística.

Otra herramienta encontrada es la desarrollada por Google, PageSpeed Insights. (Google, 2015). Su primera versión fue liberada al mercado el 2015, sin embargo, continúa siendo desarrollada y la última actualización fue en octubre de 2022.

Esta herramienta tiene varias similitudes con la herramienta planteada en la solución, considerando que ambas presentan una manera automática de evaluar una página web. Con una interfaz simple en primera instancia, solicita al usuario la URL de la página a evaluar. Luego de realizar la evaluación, entrega al usuario una lista detallada de los problemas encontrados respecto a 4 temas principales: Rendimiento, Accesibilidad, Buenas Prácticas y SEO.

Para cada uno de los temas entrega una lista detallada de problemas encontrados, como por ejemplo mencionar que se encontró imágenes en formato JPG y PNG y recomienda cambiarlas por imágenes en formatos más nuevos como WEB que permiten una mejor compresión y un menor tiempo de carga. La herramienta considera que una página web con buen rendimiento es una que presenta buenos tiempos de carga y enfoca el análisis

realizado en determinar cómo disminuirlos. La lista detallada que entrega puede resultar abrumadora para un usuario poco entendido con el tema, lo que puede generar que la herramienta deba ser usada por un público con conocimiento de causa, lo que disminuye su alcance.

Si bien la herramienta desarrollada por Google es de tipo automático, no presenta, a juicio del autor de este escrito, la mejor interfaz para usuarios no comprendidos en el tema de la evaluación heurística. Lo anterior genera una dificultad al momento de querer utilizar la herramienta si se trata de una persona común sin conocimientos de desarrollo de páginas web o de evaluación heurística.

Considerando lo planteado respecto a las 2 herramientas encontradas, se sigue identificando la necesidad de automatizar una herramienta que realice una evaluación heurística que presente una interfaz lo suficientemente simple y que genere resultados que cualquier persona sea capaz de comprender, sea que se trate de un desarrollador o de un usuario común y corriente sin más conocimientos que el saber navegar por internet.

Otro ejemplo de guía para realizar una evaluación heurística es la presentada por David Travis (Travis, 2016). Esta guía se presenta como un ejemplo concreto de evaluación, con un gran número de puntos que es necesario que el usuario evalúe en la página y sobre los cuáles debe poder identificar una nota que será ingresada en un documento Excel.

El documento está compuesto por los siguientes temas: Página Principal, Orientación a Tareas y Funcionalidad, Navegabilidad, Formularios y Entrada de Datos, Confianza y Credibilidad, Calidad de Contenido y Escritura, Diagramación y Diseño Gráfico, Búsquedas, Ayuda, Retroalimentación y Recuperación de Errores. Se identifica que estos conceptos están fuertemente relacionados con las heurísticas de Nielsen.

Para poder utilizar la guía del doctor Travis, es necesario contestar 247 preguntas sobre los temas mencionados. Esto presenta inmediatamente un proceso que no todo usuario va a estar dispuesto a realizar para obtener comentarios sobre la página web seleccionada. Se identifica que, si bien esta guía es bien completa, no resulta amigable de usar, más bien resulta en primera instancia intimidante por la cantidad de preguntas y conceptos que se debe entender y manejar.

Se llega a la conclusión de que si bien existen soluciones en el mundo actualmente que realicen de alguna manera u otra lo que intentará realizar la herramienta planteada en este escrito, se identifica que los alcances de estas soluciones y la manera en que deben ser utilizadas presenta más dificultades que soluciones. Además de lo anterior, las herramientas evaluadas están enfocadas y se basan en convenciones internacionales de evaluación, mientras que la herramienta planteada está pensada y se basa en lo obtenido para el mercado chileno de páginas web.

Debido a lo anterior es que se considera que la herramienta planteada puede llegar a solucionar el problema identificado, dado que está pensada para un mercado más acotado y por lo tanto se puede diseñar de tal manera que cumpla con su objetivo de mejor manera

y no caiga en la problemática que se encuentra en las otras soluciones en el mercado: tener un estudio que requiera de conocimiento previo, que sea complicado de utilizar o que presente una manera muy engorrosa de ser usado.

CAPÍTULO 3: PROPUESTA DE SOLUCION

3.1 IMPLEMENTACIÓN

El desarrollo de la herramienta propuesta se llevará a cabo utilizando el lenguaje Python, junto con librerías que ayudarán a configurar las distintas funcionalidades. La librería más importante será Selenium (ThoughtWorks, 2022), la que será utilizada para realizar un tipo de *webscrapping* (Amos, 2022). Con este método es posible reconocer las distintas partes y secciones de una página web, analizando el código html.

Se decide realizar la implementación de 3 programas o códigos. El primero de estos corresponde al evaluador automático de páginas web como tal, que recibe una URL del usuario y entrega resultados luego de una breve carga en formato web. El 2do código es el analizador de las 50 páginas web chilenas más visitadas. Este programa se encargará de poblar y actualizar la base de datos con datos relevantes obtenidos de las 50 páginas analizadas. Estos datos serán usados para crear una parametría que será usada de la mano con la obtenida de las heurísticas de Nielsen y las convenciones chilenas al momento de realizar la evaluación automática. El 3er script comprende la creación de los parámetros en base a lo obtenido con el analizador de las 50 páginas.

Junto con estos scripts escritos en Python, se incluye la configuración de una base de datos, la cual dispondrá de tablas configuradas para almacenar todo lo necesario para realizar tanto el análisis como la evaluación automática.

Se presenta el código desarrollado en el Anexo 3.

3.2 BASE DE DATOS

La configuración de esta base de datos se fue realizando de manera gradual durante el desarrollo de la herramienta, creando y modificando tablas según fuera necesario para añadir nuevas funcionalidades o modificar las ya creadas.

A continuación, se presenta una captura de la base de datos con sus tablas y una versión resumida de cada tabla para mostrar el formato y ejemplos de datos.

3.2.1 MODELO BD

La base de datos está compuesta por 11 tablas, las que almacenan datos obtenidos de ambos procesos, el analizador y el evaluador.

<table border="1"> <thead> <tr> <th colspan="2">dbo.color</th> </tr> </thead> <tbody> <tr> <td>url</td> <td>varchar(50)</td> </tr> <tr> <td>background</td> <td>varchar(30)</td> </tr> <tr> <td>letra</td> <td>varchar(30)</td> </tr> </tbody> </table>	dbo.color		url	varchar(50)	background	varchar(30)	letra	varchar(30)	<table border="1"> <thead> <tr> <th colspan="2">dbo.dimensiones</th> </tr> </thead> <tbody> <tr> <td>url</td> <td>varchar(50)</td> </tr> <tr> <td>page_height</td> <td>varchar(10)</td> </tr> <tr> <td>page_width</td> <td>varchar(10)</td> </tr> <tr> <td>header_height</td> <td>varchar(10)</td> </tr> <tr> <td>header_width</td> <td>varchar(10)</td> </tr> <tr> <td>footer_height</td> <td>varchar(10)</td> </tr> <tr> <td>footer_width</td> <td>varchar(10)</td> </tr> <tr> <td>search_height</td> <td>varchar(10)</td> </tr> <tr> <td>search_width</td> <td>varchar(10)</td> </tr> <tr> <td>logo_height</td> <td>varchar(10)</td> </tr> <tr> <td>logo_width</td> <td>varchar(10)</td> </tr> </tbody> </table>	dbo.dimensiones		url	varchar(50)	page_height	varchar(10)	page_width	varchar(10)	header_height	varchar(10)	header_width	varchar(10)	footer_height	varchar(10)	footer_width	varchar(10)	search_height	varchar(10)	search_width	varchar(10)	logo_height	varchar(10)	logo_width	varchar(10)	<table border="1"> <thead> <tr> <th colspan="2">dbo.font_size</th> </tr> </thead> <tbody> <tr> <td>url</td> <td>varchar(50)</td> </tr> <tr> <td>font_size</td> <td>varchar(10)</td> </tr> </tbody> </table>	dbo.font_size		url	varchar(50)	font_size	varchar(10)	<table border="1"> <thead> <tr> <th colspan="2">dbo.footer</th> </tr> </thead> <tbody> <tr> <td>url</td> <td>varchar(50)</td> </tr> <tr> <td>tag</td> <td>varchar(3)</td> </tr> <tr> <td>id</td> <td>varchar(3)</td> </tr> <tr> <td>class</td> <td>varchar(3)</td> </tr> </tbody> </table>	dbo.footer		url	varchar(50)	tag	varchar(3)	id	varchar(3)	class	varchar(3)
dbo.color																																																			
url	varchar(50)																																																		
background	varchar(30)																																																		
letra	varchar(30)																																																		
dbo.dimensiones																																																			
url	varchar(50)																																																		
page_height	varchar(10)																																																		
page_width	varchar(10)																																																		
header_height	varchar(10)																																																		
header_width	varchar(10)																																																		
footer_height	varchar(10)																																																		
footer_width	varchar(10)																																																		
search_height	varchar(10)																																																		
search_width	varchar(10)																																																		
logo_height	varchar(10)																																																		
logo_width	varchar(10)																																																		
dbo.font_size																																																			
url	varchar(50)																																																		
font_size	varchar(10)																																																		
dbo.footer																																																			
url	varchar(50)																																																		
tag	varchar(3)																																																		
id	varchar(3)																																																		
class	varchar(3)																																																		
<table border="1"> <thead> <tr> <th colspan="2">dbo.header</th> </tr> </thead> <tbody> <tr> <td>url</td> <td>varchar(50)</td> </tr> <tr> <td>tag</td> <td>varchar(3)</td> </tr> <tr> <td>id</td> <td>varchar(3)</td> </tr> <tr> <td>class</td> <td>varchar(3)</td> </tr> </tbody> </table>	dbo.header		url	varchar(50)	tag	varchar(3)	id	varchar(3)	class	varchar(3)	<table border="1"> <thead> <tr> <th colspan="2">dbo.ubicacion</th> </tr> </thead> <tbody> <tr> <td>url</td> <td>varchar(50)</td> </tr> <tr> <td>header_x</td> <td>varchar(10)</td> </tr> <tr> <td>header_y</td> <td>varchar(10)</td> </tr> <tr> <td>footer_x</td> <td>varchar(10)</td> </tr> <tr> <td>footer_y</td> <td>varchar(10)</td> </tr> <tr> <td>search_x</td> <td>varchar(10)</td> </tr> <tr> <td>search_y</td> <td>varchar(10)</td> </tr> <tr> <td>logo_x</td> <td>varchar(10)</td> </tr> <tr> <td>logo_y</td> <td>varchar(10)</td> </tr> </tbody> </table>	dbo.ubicacion		url	varchar(50)	header_x	varchar(10)	header_y	varchar(10)	footer_x	varchar(10)	footer_y	varchar(10)	search_x	varchar(10)	search_y	varchar(10)	logo_x	varchar(10)	logo_y	varchar(10)	<table border="1"> <thead> <tr> <th colspan="2">dbo.logo</th> </tr> </thead> <tbody> <tr> <td>url</td> <td>varchar(50)</td> </tr> <tr> <td>tag</td> <td>varchar(3)</td> </tr> <tr> <td>id</td> <td>varchar(3)</td> </tr> <tr> <td>class</td> <td>varchar(3)</td> </tr> </tbody> </table>	dbo.logo		url	varchar(50)	tag	varchar(3)	id	varchar(3)	class	varchar(3)									
dbo.header																																																			
url	varchar(50)																																																		
tag	varchar(3)																																																		
id	varchar(3)																																																		
class	varchar(3)																																																		
dbo.ubicacion																																																			
url	varchar(50)																																																		
header_x	varchar(10)																																																		
header_y	varchar(10)																																																		
footer_x	varchar(10)																																																		
footer_y	varchar(10)																																																		
search_x	varchar(10)																																																		
search_y	varchar(10)																																																		
logo_x	varchar(10)																																																		
logo_y	varchar(10)																																																		
dbo.logo																																																			
url	varchar(50)																																																		
tag	varchar(3)																																																		
id	varchar(3)																																																		
class	varchar(3)																																																		
<table border="1"> <thead> <tr> <th colspan="2">dbo.resultados</th> </tr> </thead> <tbody> <tr> <td>item</td> <td>varchar(20)</td> </tr> <tr> <td>resultado</td> <td>varchar(50)</td> </tr> <tr> <td>comentario</td> <td>varchar(500)</td> </tr> </tbody> </table>	dbo.resultados		item	varchar(20)	resultado	varchar(50)	comentario	varchar(500)	<table border="1"> <thead> <tr> <th colspan="2">dbo.parametros</th> </tr> </thead> <tbody> <tr> <td>item</td> <td>varchar(20)</td> </tr> <tr> <td>metodo</td> <td>varchar(30)</td> </tr> </tbody> </table>	dbo.parametros		item	varchar(20)	metodo	varchar(30)	<table border="1"> <thead> <tr> <th colspan="2">dbo.loadtime</th> </tr> </thead> <tbody> <tr> <td>url</td> <td>varchar(50)</td> </tr> <tr> <td>tiempo</td> <td>varchar(10)</td> </tr> </tbody> </table>	dbo.loadtime		url	varchar(50)	tiempo	varchar(10)	<table border="1"> <thead> <tr> <th colspan="2">dbo.searchbox</th> </tr> </thead> <tbody> <tr> <td>url</td> <td>varchar(50)</td> </tr> <tr> <td>tag</td> <td>varchar(3)</td> </tr> <tr> <td>id</td> <td>varchar(3)</td> </tr> <tr> <td>class</td> <td>varchar(3)</td> </tr> </tbody> </table>	dbo.searchbox		url	varchar(50)	tag	varchar(3)	id	varchar(3)	class	varchar(3)																		
dbo.resultados																																																			
item	varchar(20)																																																		
resultado	varchar(50)																																																		
comentario	varchar(500)																																																		
dbo.parametros																																																			
item	varchar(20)																																																		
metodo	varchar(30)																																																		
dbo.loadtime																																																			
url	varchar(50)																																																		
tiempo	varchar(10)																																																		
dbo.searchbox																																																			
url	varchar(50)																																																		
tag	varchar(3)																																																		
id	varchar(3)																																																		
class	varchar(3)																																																		

Figura 1: Modelo Base de Datos

Las tablas almacenan la siguiente información:

- La tabla color almacena el color del *background* y el color de la letra definidos en el cuerpo o *body* del html:

	url	background	letra
1	https://www.lun.com		
2	https://www.emol.com	rgba(255, 255, 255, 1)	rgba(0, 0, 0, 1)
3	https://www.latercera.cl	rgba(255, 255, 255, 1)	rgba(0, 0, 0, 1)
4	https://www.bancoestado.cl	rgba(0, 0, 0, 0)	rgba(51, 51, 51, 1)
5	https://www.biobiochile.cl	rgba(255, 255, 255, 1)	rgba(34, 34, 34, 1)
6	https://www.bancochile.cl	rgba(255, 255, 255, 1)	rgba(33, 34, 35, 1)
7	https://www.economicos.cl	rgba(255, 255, 255, 1)	rgba(51, 51, 51, 1)
8	https://www.santander.cl	rgba(255, 255, 255, 1)	rgba(68, 68, 68, 1)

- La tabla dimensiones almacena, en caso de encontrarse, las dimensiones de elementos buscados como el header, el footer, el cuadro de búsqueda y el logo. Además, almacena las dimensiones totales de la página web:

	url	page_height	page_width	header_height	header_width	footer_height	footer_width	search_height	search_width	logo_height	logo_width
1	https://www.lun.com	962	1920	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
2	https://www.emol.com	6536	960	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
3	https://www.latercera.cl	28175	1903	0	1903	692	1903	40	50	40	0
4	https://www.bancoestado.cl	3194	1903	150	1903	342	1903	NULL	NULL	36	188
5	https://www.biobiochile.cl	19394	1903	678	1903	686	1903	NULL	NULL	41	140
6	https://www.bancochile.cl	3834	1903	64	1903	423	1903	40	0	NULL	NULL
7	https://www.economicos.cl	962	1920	101	1920	75	1920	NULL	NULL	NULL	NULL
8	https://www.santander.cl	6452	1903	87	1903	184	1903	NULL	NULL	36	180

- La table font_size almacena el tamaño de la letra definido en el cuerpo o *body* del html:

	url	font_size
1	https://www.lun.com	
2	https://www.emol.com	16px
3	https://www.latercera.cl	16px
4	https://www.bancoestado.cl	13px
5	https://www.biobiochile.cl	16px
6	https://www.bancochile.cl	14px
7	https://www.economicos.cl	12.16px
8	https://www.santander.cl	16px

- La tabla footer almacena 0 o 1 dependiendo de si se encuentra el elemento, buscándolo por *tag_name*, *id* o *class_name*:

	url	tag	id	class
1	https://www.lun.com	0	0	0
2	https://www.emol.com	0	0	0
3	https://www.latercera.cl	1	0	0
4	https://www.bancoestado.cl	0	1	0
5	https://www.biobiochile.cl	1	0	0
6	https://www.bancochile.cl	0	1	0
7	https://www.economicos.cl	1	0	0
8	https://www.santander.cl	1	0	0

- La tabla header almacena 0 o 1 dependiendo de si se encuentra el elemento, buscándolo por *tag_name*, *id* o *class_name*:

	url	tag	id	class
1	https://www.lun.com	0	0	0
2	https://www.emol.com	0	0	0
3	https://www.latercera.cl	1	0	0
4	https://www.bancoestado.cl	0	1	0
5	https://www.biobiochile.cl	1	0	0
6	https://www.bancochile.cl	1	0	0
7	https://www.economicos.cl	1	0	0
8	https://www.santander.cl	1	0	0

- La tabla loadtime almacena los tiempos de carga de cada página visitada:

	url	tiempo
1	https://www.lun.com	2.231
2	https://www.emol.com	2.935
3	https://www.latercera.cl	7.911
4	https://www.bancoestado.cl	2.029
5	https://www.biobiochile.cl	4.702
6	https://www.bancochile.cl	1.734
7	https://www.economicos.cl	1.046
8	https://www.santander.cl	1.005

- La tabla logo almacena 0 o 1 dependiendo de si se encuentra el elemento, buscándolo por *tag_name*, *id* o *class_name*:

	url	tag	id	class
1	https://www.lun.com	0	0	0
2	https://www.emol.com	0	0	0
3	https://www.latercera.cl	0	0	1
4	https://www.bancoestado.cl	0	1	0
5	https://www.biobiochile.cl	0	0	1
6	https://www.bancochile.cl	0	0	0
7	https://www.economicos.cl	0	0	0
8	https://www.santander.cl	0	0	1

- La tabla searchbox almacena 0 o 1 dependiendo de si se encuentra el elemento, buscándolo por *tag_name*, *id* o *class_name*:

	url	tag	id	class
1	https://www.lun.com	0	0	0
2	https://www.emol.com	0	0	0
3	https://www.latercera.cl	0	0	1
4	https://www.bancoestado.cl	0	0	0
5	https://www.biobiochile.cl	0	0	0
6	https://www.bancochile.cl	0	1	0
7	https://www.economicos.cl	0	0	0
8	https://www.santander.cl	0	0	0

- La tabla ubicación almacena las coordenadas de los elementos encontrados:

	url	header_x	header_y	footer_x	footer_y	search_x	search_y	logo_x	logo_y
1	https://www.lun.com	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
2	https://www.emol.com	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
3	https://www.latercera.cl	0	40	0	27483	125	0	175	100
4	https://www.bancoestado.cl	0	0	0	2838	NULL	NULL	482	61
5	https://www.biobiochile.cl	0	0	0	18708	NULL	NULL	412	3188
6	https://www.bancochile.cl	0	40	0	3411	566	52	NULL	NULL
7	https://www.economicos.cl	0	0	0	760	NULL	NULL	NULL	NULL
8	https://www.santander.cl	0	0	0	6268	NULL	NULL	398	41

- La tabla de parámetros se obtiene en base los datos almacenados en las tablas anteriores, usando un script extra. Los ítems corresponden a cada punto que será discutido y evaluado por el evaluador automático. El método es en la mayoría de los casos el resultado esperado. En casos como la altura y el ancho, o las coordenadas de los elementos, corresponde a la media de los datos obtenidos de las tablas anteriores:

	ítem	metodo
1	header	tag
2	footer	tag
3	searchbox	class
4	logo	class
5	loadtime	2,67
6	background	rgba(255, 255, 255, 1)
7	letra	rgba(0, 0, 0, 1)
8	font_size	16px
9	page_height	7.028
10	page_width	1.884
11	search_height	33
12	header_height	136
13	header_width	1.795
14	header_x	28
15	header_y	18
16	footer_height	400
17	footer_width	1.839
18	footer_x	9
19	footer_y	6.695
20	search_width	219
21	search_x	480
22	search_y	84
23	logo_height	55
24	logo_width	190
25	logo_x	465
26	logo_y	1.153
27	header_x_m...	0
28	header_y_m...	0
29	footer_x_me...	0
30	footer_y_me...	27.483
31	search_x_m...	0
32	search_y_m...	0
33	logo_x_media	382
34	logo_y_media	1
35	font_family	Sans-serif

- La tabla resultados almacena los resultados encontrados por el evaluador automático y el comentario que se considere correcto realizar, en base a la tabla de parámetros y parámetros definidos previamente:

	item	resultado	comentario
1	url	https://www.paris.cl	NULL
2	loadtime	0,958	Tiempo de carga menor al esperado de 2,67 segundos
3	window_height	3.604	Página mas alta de lo recomendado. La altura de la ...
4	infinite	Si	La página posee infinite scrolling: 3 scrolls
5	window_width	1.903	El ancho de la página es 1.903, el ancho de la panta...
6	header	Si	Se encontró correctamente el header por el tag_name
7	header_height	101	El header es mas pequeño de lo esperado: 192.203 ...
8	header_width	1.903	NULL

Las tablas color, dimensiones, font_size, footer, header, loadtime, logo, ubicación y searchbox son llenadas por el programa referenciado anteriormente como el analizador de páginas web, que se encarga de evaluar de manera automática la muestra completa utilizada en esta investigación.

La tabla de parámetros es llenada por un programa aparte, que se encarga de realizar cálculos y comparaciones entre los datos almacenados en las otras tablas.

La tabla de resultados es la que se obtiene luego de realizar una evaluación automática de una página web específica y es la salida del sistema. Sus datos se muestran en una página web al finalizar la evaluación automática.

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN

Fue mencionado anteriormente que la solución planteada busca automatizar la evaluación heurística de páginas web. Para eso fue necesario definir qué tipo de características de una página se evaluaría y contra qué parámetro o convención se compararía para realizar un análisis.

En la sección anterior se mencionó que características se buscaría automatizar. Durante el proceso de desarrollo se priorizó algunas, mientras que otras no fueron exploradas completamente.

4.1 CARACTERÍSTICAS AUTOMATIZADAS

A continuación, se explicará las características que se decidió automatizar. Se mencionará como se obtuvo y contra qué parámetro o convención se comparó para posibilitar un comentario útil para un futuro usuario de la herramienta. En esta sección se usará el concepto de la muestra para referirse a la muestra utilizada en esta investigación, correspondiente a las 50 páginas web chilenas más visitadas. Se presentará las características separadas por temas y ordenadas en tablas.

4.1.1 ASPECTOS GENERALES

Tabla 1: Aspectos Generales

Característica	Descripción	Implementación
Tiempo de carga	Se obtuvo midiendo el tiempo desde que se realizaba la solicitud al servidor de obtener la página, hasta que el servidor respondía con el contenido y lo cargaba correctamente.	El parámetro utilizado para la comparación con el tiempo de carga específico para cada página se obtuvo calculando la media de los tiempos de carga de la Muestra, llámese el tiempo esperado.
Altura de la página	La altura de la página es el largo completo de la página, incluyendo el contenido no visible inicialmente que debe ser alcanzado mediante <i>scrolls</i> .	El parámetro utilizado para la comparación es una convención, que dicta que el tamaño máximo de una página no debe exceder las 3 pantallas completas (se debe poder realizar máximo 2 <i>scrolls</i> para acceder a todo el contenido).
Infinite scrolling	Esta característica va de la mano con la anterior ya que el resultado depende de la cantidad de	Se utiliza como parámetro, que, si la página web tiene un largo de más de 3 pantallas completas, posee <i>infinite scrolling</i> .

	pantallas completas que posea la página web.	
Ancho de la página	Esta característica se obtiene midiendo directamente el ancho de la página comparándolo con el ancho de la pantalla utilizada.	Se compara con el ancho recomendado de una página web, definido como 80% del tamaño de la ventana del explorador.
H1	Se considera buena práctica que la página contenga por lo menos un elemento con el tag html h1.	Se busca en el código un elemento con el tag h1.

4.1.2 HEADER, FOOTER, SEARCH Y LOGO

Las siguientes características se repiten para cada uno de los 4 elementos de la página web: *header*, *footer*, *search* y *logo*.

Tabla 2: Header, Footer, Search y Logo

Característica	Descripción	Implementación
Se encontró el elemento	Este ítem indica si fue posible encontrar los elementos en la página web.	La manera en que se buscan los elementos se obtiene de un promedio de los resultados obtenidos de la Muestra.
Altura	Indica la altura del elemento.	Se usan en conjunto para describir el tamaño del elemento, el cual se compara con la media de los tamaños obtenidos de la Muestra.
Ancho	Indica el ancho del elemento.	
Coordenada 'x'	Es la coordenada horizontal del elemento dentro del html.	Header: Se utiliza como parámetro que el <i>header</i> debe comenzar inmediatamente al inicio de la página, por lo anterior el resultado esperado es una coordenada (0,0). Footer: El parámetro utilizado indica que el <i>footer</i> debe ser el último elemento de la página, luego la coordenada "x" debe ser 0 y la
Coordenada 'y'	Es la coordenada vertical del elemento dentro del html.	

		<p>coordenada “y” debe ser igual a la resta entre la altura total de la página y la altura del elemento.</p> <p>Search: Se espera que este elemento se encuentre en la sección superior derecha de la página.</p> <p>Logo: Se espera que este elemento se encuentre en la sección superior izquierda de la página.</p>
--	--	--

4.1.3 LETRA Y COLOR

Tabla 3: Letra y Color

Característica	Descripción	Implementación
Color de fondo	Indica el color de fondo de la página web definido en el <i>body</i> del html.	Se obtiene el valor y se compara con la media de la Muestra.
Color de letra	Indica el color de la letra de la página web definido en el <i>body</i> del html.	
Tamaño de letra	Indica el tamaño de la letra de la página web definido en el <i>body</i> del html.	
Fuente del texto	Se obtiene del atributo <i>font_family</i> del <i>body</i> del html.	Se utiliza como parámetro de comparación la familia Sans-serif que es la más utilizada.
Ajuste de letra correcto	Se obtiene del atributo <i>text-size-adjust</i> del <i>body</i> del html.	Se busca la propiedad en el html y se indica cuantos ajustes de letra correctos existen. Se considera que debe ser 100%, un valor distinto se considera como incorrecto.
Ajuste de letra incorrecto		

4.1.4 ENLACES

Tabla 4: Enlaces

Característica	Descripción	Implementación
Nr de enlaces	Esta característica cumple solo la función de informar el nr de links encontrados en la página web.	Se busca todos los elementos con tag html "a" que contengan un <i>href</i> , es decir, un enlace.
Enlaces correctos	Con este ítem se explica cuántos enlaces de los encontrados cumplen con las condiciones esperadas.	Las características esperadas de todo enlace encontrado en la página es que sea de color azul y esté subrayado.
Enlaces incorrectos		
Enlaces subrayados	Indica cuantos de los enlaces encontrados estaban correctamente subrayados.	Se busca en el enlace el atributo " <i>text-decoration</i> ", si es igual a " <i>underline</i> " se dice que está correctamente subrayado.
Enlaces azules	Indica cuántos de los enlaces encontrados tenían el color correcto.	Se compara el color del enlace obtenido del atributo " <i>color</i> " con el color rgb azul, mediante una comparación delta e de los valores rgb. Si el valor obtenido es menor a 10 se dice que es "igual" al azul.
Enlaces suficientemente distintos	Indica cuántos de los enlaces encontrados tienen un color lo suficientemente parecido al azul utilizado en la comparación.	Se compara el valor rgb del enlace obtenido con el azul con delta e. Si el valor obtenido según la métrica utilizada es mayor a 10, se realiza nuevamente la comparación con un valor rgb obtenido de la media de la Muestra.

4.1.5 PUBLICIDAD Y MEDIA

Tabla 5: Publicidad y Media

Característica	Descripción	Implementación
Nr de videos	Indica la cantidad de videos con reproducción automática encontrados en la página web.	Se busca elementos del html con el tag <i>"videos"</i> que tengan el atributo <i>"autoplay"</i> .
Nr de audios	Indica la cantidad de elementos de audio con reproducción automática encontrados en la página web.	Se busca elementos del html con el tag <i>"audio"</i> que tengan el atributo <i>"autoplay"</i> .
Nr de anuncios	Indica la cantidad de anuncios encontrados en la página web.	Se busca elementos del html con el tag <i>"iframe"</i> cuyo atributo <i>"display"</i> sea distinto a <i>"none"</i> .

Este fue el set de heurísticas que fue posible automatizar. Dentro de las mencionadas en capítulos anteriores se encontraba un gran número de características o de heurísticas de Nielsen, que requerían de un ojo humano para ser identificadas y evaluadas.

Lo anterior se debe principalmente a la necesidad de identificar un tipo de contenido específico en la página, especialmente texto. Dado que identificar texto necesita de herramientas más complejas para ser llevado a cabo, se determinó que se escapaba del objetivo principal de esta investigación.

Las características que si fueron automatizadas y fueron finalmente usadas en el evaluador automático de páginas web corresponden a los elementos de toda página web que fue posible encontrar con la herramienta utilizada.

Fue mencionado como un objetivo secundario el crear un set de convenciones en base a la muestra. Este resultado se puede evidenciar en la tabla de parámetros, dado que un gran número de parámetros fue obtenido del análisis de la muestra respecto a las convenciones consideradas previamente.

De este análisis fue posible calcular las medias y obtener de esta manera en algunos casos un tamaño promedio esperado para ciertos elementos. Por otro lado, fue posible identificar valores esperados para un número de características. Si bien se tenía un valor esperado dado por las heurísticas de Nielsen, se consideró que obtener un valor gracias a la evaluación de la Muestra sería más significativo y lograría mejor el objetivo principal.

4.2 RESULTADOS

A continuación, se presentan resultados obtenidos de la misma manera en las que los podrá ver un futuro usuario de la herramienta. Se debe mencionar que los resultados dependen de la página web evaluada. Los resultados que se mostrarán a continuación corresponden a los obtenidos luego de evaluar la página con URL: <https://www.paris.cl>.

Tabla 6: Resultados obtenidos para <https://www.paris.cl/>

Glosa	Medición	Comentario
Tiempo de carga	0,958	Tiempo de carga menor al esperado de 2,67 segundos
Altura de la página	3.604	Página más alta de lo recomendado. La altura de la página es 3.604 pixeles y La altura recomendada es de no más de 3 pantallas completas. La altura de la pantalla utilizada es de 1.056 pixeles
Infinite scrolling	Si	La página posee infinite scrolling: 3 scrolls
Ancho de la página	1.903	El ancho de la página es 1.903, el ancho de la pantalla es 1.936 y el ancho recomendado de la página es 1.548
Se encontró h1	Sí	Se encontró correctamente al menos un h1: NAVIDAD PARIS
Se encontró header	Si	Se encontró correctamente el header por el tag_name
Altura del header	101	El header es más pequeño de lo esperado: 192.203 < 244.120
Ancho del header	1.903	
Coordenada 'x' del header	0	El header se encuentra en la posición esperada: (0, 0) == (0,0)
Coordenada 'y' del header	0	
Se encontró footer	Si	Se encontró correctamente el footer por el tag_name
Altura del footer	737	El footer es más grande de lo esperado: 1.402.511 > 735.600
Ancho del footer	1.903	

Coordenada 'x' del footer	0	El footer no se encuentra en la posición esperada, dado que se espera que su coordenada en 'x' sea 0 y su coordenada en 'y' sea la resta entre la altura de la página de 14.043 pixeles y su altura de 737 pixeles: 13.306
Coordenada 'y' del footer	13.292	
Se encontró search	No	No se encontró el search buscando por class_name
Se encontró logo	Si	Se encontró correctamente el logo por el class_name
Altura del logo	74	Logo más grande de lo esperado: 14.652 > 14.400
Ancho del logo	198	
Coordenada 'x' del logo	15	El logo no se encuentra en la posición esperada (Superior izquierda)
Coordenada 'y' del logo	27	
Color de fondo	rgba(255, 255, 255, 1)	El color de fondo es igual al esperado: rgba(255, 255, 255, 1)
Color de la letra	rgba(26, 26, 26, 1)	El color de la letra no es igual al esperado: rgba(0, 0, 0, 1)
Tamaño de la letra	10px	El tamaño de la letra es menor al esperado: 16px
Fuente del texto	"Open Sans", sans-serif	Se encontró la fuente de texto esperada dentro de la familia de fuentes
Ajuste de letra correcto	text-size-adjust: 100%	Se encontró 1 propiedad que fija bien el tamaño de la letra: text-size-adjust: 100%
Ajuste de letra incorrecto		No se encontró propiedades que fijen mal el tamaño de la letra
Nr de links	225	La página posee 225 links
Links correctos	0	Ningún link cumple con lo esperado
Links incorrectos	225	
Links subrayados	0	El nr de links subrayados es: 0

Links azules	0	Los links que cumplen con el color azul son: 0
Links suficientemente distintos	186	Los links que tienen un color distinto al azul pero suficientemente distinto al color de la letra son: 186
Nr de videos	0	No se detectó videos que se reprodujeran automáticamente
Nr de audios	0	No se detectó pistas de audio que se reprodujeran automáticamente
Nr de anuncios	0	No se encontró anuncios en la página

Se puede evidenciar en esta tabla los resultados obtenidos por un análisis de una página web en específico, utilizando la herramienta desarrollada en esta investigación. En la primera columna se menciona el tipo de característica que fue buscada, en la segunda columna se presenta el dato medido y en la tercera columna se presenta el resultado de la evaluación en base al dato medido y a los parámetros utilizados.

Estos resultados también se pueden apreciar en el Anexo 3 RESULTADO GRÁFICO EVALUACIÓN.

4.3 CONVENCION DE DISEÑO DE PÁGINAS WEB CHILENAS

Como fue mencionado en secciones anteriores, el objetivo secundario de este estudio era encontrar un set de convenciones chilenas que derivara de un estudio realizado a las 50 páginas chilenas más visitadas. Los resultados de esta evaluación se pueden ver a continuación:

Tabla 7: Resultados obtenidos para el análisis de la muestra (objetivo secundario)

Característica	Glosa	Parámetro
Header	Método de búsqueda	tag
	Altura del header	136
	Ancho del header	1.795
	Coordenada 'x' (Promedio)	28
	Coordenada 'y' (Promedio)	18
	Coordenada 'x' (Media)	0
	Coordenada 'y' (Media)	0

Footer	Método de búsqueda	tag
	Altura del footer	400
	Ancho del footer	1.839
	Coordenada 'x' (Promedio)	9
	Coordenada 'y' (Promedio)	6.695
	Coordenada 'x' (Media)	0
	Coordenada 'y' (Media)	27.483
Barra de búsqueda	Método de búsqueda	class
	Altura de la barra de búsqueda	33
	Ancho de la barra de búsqueda	219
	Coordenada 'x' (Promedio)	480
	Coordenada 'y' (Promedio)	84
	Coordenada 'x' (Media)	0
	Coordenada 'y' (Media)	0
Logo	Método de búsqueda	class
	Altura del logo	55
	Ancho del logo	190
	Coordenada 'x' (Promedio)	465
	Coordenada 'y' (Promedio)	1.153
	Coordenada 'x' (Media)	382
	Coordenada 'y' (Media)	1
Tiempo de carga y dimensiones generales	Tiempo de carga	2,67
	Altura de la página	7.028

	Ancho de la página	1.884
Letra y Color	Color de fondo	rgba(255, 255, 255, 1)
	Color de la letra	rgba(0, 0, 0, 1)
	Tamaño de la letra	16px
	Fuente del texto	Sans-serif

En esta tabla se observa el set de convenciones que se obtuvo mediante la evaluación de las 50 páginas chilenas más visitadas. Este set de convenciones se usa para el análisis desarrollado en esta investigación, junto con las convenciones de Nielsen que fueron automatizadas. Se puede apreciar estos resultados en el Anexo 4 RESULTADO GRÁFICO PARÁMETROS.

De la tabla de resultados obtenidos se desprende 6 categorías, dentro de las cuáles se encuentran los distintos parámetros que fueron obtenidos del análisis de la muestra.

Para *Header*, *Footer*, Logo y Barra de Búsqueda se observan los mismos parámetros, cada uno con los resultados específicos en cada categoría. En primer lugar, se observa el método de búsqueda. Se consideró necesario mostrar y utilizar este parámetro dado que dicta como será buscada la característica en la página web analizada. Las opciones consideradas en esta búsqueda fueron por *tag-name*, *class-name* e *id*. Se busca de 3 maneras dado que se encontró de estas 3 formas en la mayoría de las páginas de la muestra. Dado que los resultados eran variados, para el análisis de la muestra se buscó de las 3 maneras y el parámetro se obtuvo sacando la media de lo obtenido.

Los parámetros de altura, ancho y las coordenadas fueron obtenidos mediante el método de búsqueda mencionado anteriormente. Para estos se calculó el promedio de los resultados, que son los que se muestran en la tabla. Para las coordenadas también se utiliza la media de la muestra, dado que se considera un mejor dato para realizar la comparación.

Para los parámetros de tiempo de carga y dimensiones generales de la página se obtuvo el promedio de los datos obtenidos del análisis de la muestra, estando el tiempo en segundos y la altura y el ancho en pixeles.

En la categoría de letra y color se calcula la media de los resultados obtenidos.

Si bien se ha usado el trabajo de Nielsen como una base conceptual en este estudio, se puede identificar que los resultados obtenidos no contienen todos los lineamientos planteados por Nielsen, que son mencionados en el segundo capítulo de este escrito (CAPÍTULO 2: MARCO CONCEPTUAL).

Lo anterior se debe principalmente a que los lineamientos de Nielsen consideran una mirada humana subjetiva sobre una página web, además de la lectura de texto y la

comprensión del contenido general de la página. Dado que la búsqueda de este tipo de características no estaba considerada en el alcance de esta investigación es que se dejó estos lineamientos fuera.

Se puede observar que se puede generar un set de convenciones respecto a la Muestra de carácter principalmente organizacional, considerando la búsqueda del *header*, *footer*, barra de búsqueda y logo. Para cada uno de estos elementos se obtiene el método con el cual están implementados, el tamaño y las coordenadas dentro de la página. De la mano con estas características se obtiene también datos relevantes para el tiempo de carga y las dimensiones generales de una página, así como también las características que tienen que ver con la letra y el color.

Esta investigación buscaba automatizar la evaluación heurística de páginas web. Respecto a los datos obtenidos del análisis de las 50 páginas chilenas más visitadas, se considera que se obtiene un set de parámetros que permiten realizar un análisis a cualquier página chilena de buena manera y que sumados a los parámetros utilizados en la herramienta que analiza paginas particulares, permite que la herramienta entregue una respuesta satisfactoria al usuario.

CAPÍTULO 5: CONCLUSIONES

El análisis de usabilidad de páginas web es un campo amplio, que permite evaluar un sinnúmero de características, pasando de lo más básico y concreto a lo más amplio y subjetivo, como lo son por ejemplo el color de los enlaces y el sentimiento que causa al usuario el navegar por una página, respectivamente.

Debido a lo planteado anteriormente es que se han dispuesto un número de autores, contando los citados en este escrito, a descubrir las mejores prácticas a ser evaluadas en un posible análisis. Este hipotético set de reglas puede resultar abrumador para un analizador, por lo que se ha intentado en esta investigación limitar el entendimiento que necesite tener tal analizador para poder realizar un análisis correcto y certero.

Se planteó como el objetivo principal de esta investigación el desarrollo de una herramienta que fuera capaz de brindar información a un posible usuario de manera rápida y eficaz, mediante la automatización del proceso de evaluación.

El alcance de esta investigación comprendía el desarrollo de una herramienta simple y eficaz, que disponibilizará información útil al usuario y que permitiera un entendimiento de las métricas utilizadas y los resultados entregados para cada una, cumpliendo con el objetivo principal del trabajo.

Se debe considerar que no es posible la automatización de todas las heurísticas y convenciones exploradas en el escrito. Esto se debe a que el método investigado y promocionado por Nielsen y los otros autores citados comprende un entendimiento del correcto funcionamiento de una página web. Esto, a su vez, genera que las heurísticas y las convenciones exploradas no presenten una línea directa a ser automatizadas.

En segundo lugar, la persona que desee utilizar estos lineamientos para realizar una evaluación debe ser capaz de aplicarlos y saber de qué forma buscar y encontrar los distintos elementos de una página web. Esto genera una dificultad real al momento de intentar automatizar estos lineamientos, dado que dependen en su mayoría de un ojo humano subjetivo, capaz de leer y comprender texto y de identificar partes distintas de una página web en relación con el contenido que muestran.

Junto con lo anterior, se fue descubriendo durante el desarrollo de la herramienta las limitaciones que esta presentaría. Algunas de estas limitaciones tienen relación directa con lo mencionado anteriormente; se tiene un set de heurísticas que no presentan una posibilidad real de ser automatizadas. Se tratan de esta manera estos lineamientos dado que como fue explorado y explicado en capítulos anteriores, muchos de los lineamientos de Nielsen o de las convenciones que se encuentran a nivel internacional dependen de la lectura y análisis de texto, aspecto que no estaba considerado dado que se escapa del alcance de la investigación.

Otro tipo de limitaciones se dieron a conocer a medida que se utilizaba y exploraba el lenguaje y la librería usados. En un principio se puede haber considerado la automatización

de todas las características posibles, pero rápidamente fue necesario desechar un número de ellas dado que el sistema no sería capaz de llevar su evaluación a cabo.

El principal resultado obtenido fue una página web, la cual se construye en base a lo obtenido en el análisis y que muestra todo el contenido de la tabla de resultados, con explicaciones relevantes para cada ítem explorado, así como también una serie de textos y contenido que cumple la función de explicar y orientar al usuario en el uso y entendimiento de la herramienta y los resultados obtenidos.

Los resultados obtenidos para un análisis de una página web utilizando la herramienta desarrollada en esta investigación se pueden apreciar en el Anexo 3 (3 RESULTADO GRÁFICO EVALUACIÓN). Se puede evidenciar en estas capturas que fue posible la automatización de un gran número de características de una página web. Las distintas características que terminaron siendo automatizadas pasaron por un largo proceso de diferenciación y catalogación, para finalmente terminar con un set de elementos que permitiera un análisis correcto y útil.

El resultado gráfico para el objetivo secundario, que corresponde al análisis de las 50 páginas chilenas más visitadas y la posterior obtención de un set de parámetros a partir de esos resultados, se puede evidenciar en el Anexo 4 (4 RESULTADO GRÁFICO PARÁMETROS). Se intenta mostrar los parámetros obtenidos de la manera más limpia posible, con explicaciones donde es necesario para comprender unidades de medida y formas de obtención de los distintos elementos.

Según se puede evidenciar en la tabla de resultados generales (4.2 RESULTADOS) y en la tabla de resultados del análisis de la muestra (Tabla 7: Resultados obtenidos para el análisis de la muestra (objetivo secundario)), se ha cumplido tanto con el objetivo principal como con el objetivo secundario planteados en esta investigación.

Al momento de realizar la investigación previa al desarrollo, se notó rápidamente que el alcance de la herramienta podía ser muy amplio si se quisiera. Entre las características que se podrían haber incluido se encontraba un gran número de heurísticas y convenciones que involucraban la lectura de texto. La inclusión de una funcionalidad capaz de identificar texto fue considerada, pero se notó rápidamente que se escaparía de los alcances de esta investigación dada la tecnología y las técnicas que sería necesario utilizar.

En las primeras secciones de este escrito, se plantea que una de las principales razones por las que se desea implementar esta solución es disponibilizar una herramienta que facilite el proceso de análisis de usabilidad de una página web. En este sentido se puede mencionar que lo analizado y construido durante el desarrollo de esta investigación puede resultar atractivo para desarrolladores que no tengan mucho conocimiento específico sobre el análisis de usabilidad, pero que si quieren usar sus conceptos para mejorar la experiencia de usuario y el diseño de su página web.

Un desarrollador que tenga nociones de análisis de usabilidad podrá utilizar esta herramienta para reforzar sus conocimientos y acuñar sus capacidades, logrando ser un profesional mejor capacitado para el desarrollo de páginas web.

Dado que esta herramienta fue pensada y desarrollada en base a una muestra compuesta por páginas web chilenas, es que puede resultar más atractiva para desarrolladores chilenos con clientes chilenos, que puedan solicitar páginas del mismo estilo que las evaluadas pero que deseen que sean mejores y más efectivas. Al usar esta herramienta durante el desarrollo de una página web chilena será posible identificar inmediatamente que tipo de aspectos deben ser corregidos o cambiados, para así mejorar su rendimiento.

No obstante lo anterior, dado que las convenciones de desarrollo planteadas por Nielsen son de carácter internacional, es que la herramienta también logra un alcance a páginas fuera de Chile, lo que posibilita su uso con cualquier página web que se desee analizar.

Pasando a otro tema relevante, como el impacto que tuvo el trabajo de Nielsen en esta investigación, es que se puede discutir que conceptualmente fue una pieza fundamental. Sin las bases trabajadas y presentadas por Nielsen toda idea de evaluación de página web tendría, probablemente, un fundamento completamente distinto que no habría posibilitado la investigación realizada.

Se menciona en este escrito que la evaluación propuesta por Nielsen comprende la necesidad de un encargado y de un set de evaluadores. El encargado debe ser quien organice la evaluación y proponga las condiciones a los evaluadores. Debe manejar los conceptos y las heurísticas que necesita evaluar en la página y debe ser capaz de explicar a los evaluadores que deben buscar en la página. Los evaluadores por su parte deben ser personas naturales, con o sin conocimiento previo de este proceso o de algún concepto de evaluación heurística, que estarán encargados de llevar a cabo la evaluación de la página en base a las condiciones establecidas por el encargado.

El proceso mencionado anteriormente comprende una serie de pasos y capacidades, además de necesitar de un número de personas, debiendo estar una de ellas necesariamente calificada y entrenada, para ser llevado a cabo. Comparando este proceso con el logrado en este escrito se puede evidenciar una clara mejoría en términos de esfuerzos necesarios para llevarlo a cabo. La herramienta desarrollada en esta investigación no necesita de un grupo de evaluadores que se deban tomar su tiempo para navegar una página y lograr extraer conclusiones sobre su funcionamiento. La solución posibilita una evaluación heurística en base a parámetros preestablecidos y es manejada por un solo individuo, que puede o no tener conocimientos sobre evaluación.

Por otro lado, comprende un mejor uso de tiempo, dado que posibilita una evaluación con una duración total de pocos minutos, desde que se ingresa la URL a ser evaluada hasta que se obtiene y lee los resultados. Al contrario de una evaluación heurística como la planteada por Nielsen, que depende de una mirada subjetiva de un usuario intentando realizar juicios de valor sobre las distintas características de la página, esta solución evalúa la página sin

necesidad de saber que está buscando el evaluador y disponibiliza información relevante sobre todos los temas que puedan estar configurados en los parámetros.

5.1 RECOMENDACIONES

Se identifica que, si bien la solución presenta un análisis para un variado número de características, se puede expandir y completar, para así lograr una herramienta capaz de identificar todas las secciones de una página web y así entregar un análisis verdaderamente completo y exhaustivo. La solución planteada en esta investigación cumple con lo esperado, pero lo anterior no deja fuera la posibilidad de su mejora y perfeccionamiento.

Tomando en cuenta lo mencionado anteriormente, se puede recomendar una serie de ideas o lineamientos a seguir, en el caso de que se quiera expandir el alcance de la herramienta. En primer lugar, se recomienda un manejo elevado del lenguaje de programación que se desee utilizar, para de esta manera no tener que malgastar tiempo en descubrir e investigar su uso.

Por otro lado, se recomienda el manejo de conceptos de desarrollo de software, para facilitar el entendimiento de la herramienta ya planteada y así poder mejorar tanto las funcionalidades ya desarrolladas como la organización general de código y del programa principal.

Si se desea simplemente expandir lo ya expuesto, se sugiere comenzar por aspectos similares a los ya expuestos, como lo serían distintas secciones de una página web que no han sido incluidas en este análisis.

Otra manera de expandir esta herramienta sería agregar una funcionalidad de análisis de texto, la cual como fue mencionado en reiteradas ocasiones en el escrito, puede resultar mucho más compleja que lo desarrollado. Se considera como una buena manera de complementar lo ya desarrollado, dado que un gran número de heurísticas y convenciones incluyen la lectura de texto de parte del analizador.

Se identifica que las herramientas utilizadas para el desarrollo de la solución resultan suficientes y convenientes. Lo anterior dado que el *webdriver* de *Selenium* está pensado justamente para lo que se usa en esta herramienta y presenta muchas opciones y características para expandir una posible búsqueda de características en una página web. Si bien se puede desarrollar en otro lenguaje, se considera que Python está bien catalogado actualmente y presenta todas las herramientas de un lenguaje moderno, lo que se considera como suficiente para el alcance de esta herramienta y la posible complementación que se le desee dar posteriormente a su desarrollo.

A modo de apreciaciones personales, se puede comentar que el tema presenta un gran número de características interesantes. Todo profesional que desee dedicarse al desarrollo sea de sistemas dedicados o de aplicaciones, probablemente pasará por el desarrollo de páginas web, utilizando conceptos y métodos como los explorados en esta investigación.

Considerando lo anterior es que se puede mencionar que esta investigación puede resultar útil para cualquier persona que desee dedicarse al desarrollo, dado que el desarrollo de páginas web es un campo amplio al que puede resultar difícil ingresar correctamente.

REFERENCIAS BIBLIOGRÁFICAS

- "Marianne". 2018.** 9 Principles of Good Web Design. [Online] Febrero 16, 2018. <https://www.feelingpeaky.com/9-principles-of-good-web-design/>.
- Alexa. 2020.** The Top 500 Sites on the Web. [Online]: 2020. https://www.alexa.com/topsites/category/Top/Business/Marketing_and_Advertising.
- Amos, David. 2022.** A Practical Introduction to Web Scraping in Python. [Online] Oct 17, 2022. <https://realpython.com/python-web-scraping-practical-introduction/>.
- Crestodina, Andy. Accedido: 2020.** Web Design Standards: 10 Best Practices on the Top 50 Websites. [Online] Accedido: 2020. <https://www.orbitmedia.com/blog/web-design-standards/>.
- Devaney, Erik. 2017.** 8 Guidelines for Exceptional Web Design, Usability and User Experience. [Online] Agosto 14, 2017. <https://blog.hubspot.com/blog/tabid/6307/bid/30557/6-guidelines-for-exceptional-website-design-and-usability.aspx>.
- Google. 2015.** Measure performance with PageSpeed Insights. *PageSpeed Insights*. [Online] 2015. <https://web.dev/measure/>.
- Heurix. 2021.** Heurix. [Online] 2021. <https://www.heurix.io/>.
- Lamprecht, Emil. 2022.** The Difference Between UX and UI Design – A Beginner’s Guide. [Online] Junio 7, 2022. <https://careerfoundry.com/en/blog/ux-design/the-difference-between-ux-and-ui-design-a-laymans-guide/>.
- Nielsen, Jakob. 2001.** *Homepage Usability: 50 Websites Deconstructed*. s.l. : New Riders Publishing, 2001.
- . **2002.** Top 10 Guidelines for Homepage Usability. [Online] Mayo 11, 2002. <https://www.nngroup.com/articles/top-ten-guidelines-for-homepage-usability/>.
- . **1993.** *Usability Engineering*. 1993.
- Pypi.org. 2022.** Pypi.org. [Online] 2022. <https://pypi.org/project/pip/>.
- Smith, Alan. Accedido: 2020.** 7 Web Design Rules You Should Never Break. [Online]: 2020. <https://usabilitygeek.com/web-design-rules-you-should-never-break/>.
- ThoughtWorks. 2022.** Selenium. [Online] 2022. <https://www.selenium.dev/>.
- Toro, Manuel. 2015.** Convenciones de diseño de las páginas de inicio de los sitios Web chilenos. Tesina para optar al grado de Magister en Tecnologías de la Información, UTFSM. 2015.
- Travis, David. 2016.** 247 web usability guidelines. [Online] Abril 12, 2016. <https://www.userfocus.co.uk/resources/guidelines.html>.
- W3C. 2019.** CSS Validation Service. [Online] 2019. <https://jigsaw.w3.org/css-validator/>.
- . **2019.** Markup Validation Service. [Online] 2019. <https://validator.w3.org/>.

—. **2019.** Standards. [Online] 2019. <https://www.w3.org/standards/>.

Wordreference. 2022. Wordreference. [Online] 2022.
<https://www.wordreference.com/definicion/convencción>.

ANEXOS

1. CONVENCIONES VIGENTES

En el marco de la investigación previa al desarrollo de la herramienta propuesta, corresponde evaluar las distintas convenciones actuales respecto al diseño de páginas web.

Se mencionarán artículos que plantean un número de convenciones a nivel internacional, para dar un enfoque global a la investigación, antes de entrar a particularizar el tema y aterrizar la idea principal a convenciones centradas en páginas chilenas.

Para lo anterior es necesario, en primer lugar, definir el concepto de convención. Según (Wordreference, 2022) una convención es una: “Norma o práctica admitida por responder a precedentes o a la costumbre”. En otras palabras, el término convención se usa para describir lo que es comúnmente visto como una costumbre. Respecto al diseño de páginas web, se refiere a un set de reglas inferido del análisis de las páginas web disponibles en el mercado, sea una red social, una página informativa o un *Marketplace*.

Siendo el diseño de páginas web un ambiente principalmente subjetivo, dado que junto con las decisiones que pueda querer tomar el desarrollador se debe tomar en cuenta los deseos del cliente, quien no siempre tiene los conocimientos necesarios que las mejores decisiones requieren, es que se da la existencia de páginas web, de uso diario por miles de usuarios, que simplemente no funcionan como deberían o no logran entregar la información necesaria al usuario de la forma requerida o en el tiempo correcto. Lo anterior se debe, principalmente, a que estas páginas no están enfocadas en el usuario y su desarrollo no fue guiado por las convenciones vigentes.

Entrando en un primer ejemplo de convenciones, se tienen las mencionadas por Alan Smith en (Smith, Accedido: 2020). Alan menciona 7 convenciones en su artículo, las que se describirán a continuación (Artículo en inglés, por lo que se introducirá el tema en el nombre escrito por Alan y se traducirá y explicará en español):

1. *Avoid Legibility Issues*: Evitar Problemas de Legibilidad. “If users cannot properly read the text on your site, your message is not going through”. Si el usuario no puede leer texto como corresponde, el mensaje no será transmitido con éxito. Se explica por sí solo, si el texto no se puede leer, el usuario no entenderá lo que necesita entender. Hace hincapié en el uso correcto de contraste del texto con el fondo y recomienda usar colores oscuros para el texto con colores claros de fondo o viceversa. Además de esto, menciona que usar fuentes fáciles de leer y con tamaño adecuado evita secciones ilegibles.
2. *Apply Visual Hierarchy*: Aplicar Jerarquía Visual. “..., you can use visual hierarchy to guide users toward the selling points of your website”. Se puede usar la jerarquía visual para guiar al usuario a los puntos de interés de la página. Se refiere al orden en que un usuario procesa la información dispuesta en una página y como esto afecta el desempeño de esta. Menciona que un uso de jerarquía claro, con los

puntos de interés puestos en un lugar en que el usuario los pueda notar cuando corresponda, logran un mejor objetivo. Además de lo anterior, colores específicos para un tipo de acción (como acceder al carrito de compra o ingresar) logran también un mejor entendimiento de las funcionalidades de la página.

3. *Do Not Go Font Crazy*: No pierdas la cabeza con las fuentes. “...having too many fonts will make your site look unstructured and unprofessional...”. Tener muchas fuentes distintas hará que tu sitio se vea mal estructurado y poco profesional. Procurar tener fuentes cohesivas y profesionales es lo ideal.
4. *Have An Obvious CTA (Call to Action)*: Ten una “llamada a acción” obvia. “Your CTA serves as your selling point and will drive customer engagement”. La llamada a acción sirve como un punto de interés al momento de navegar la página y ayuda a dirigir la atención del usuario hacia lo importante. Entiéndase como CTA un botón de “Comprar Ahora” para un Marketplace, por ejemplo. Luego, lo ideal es que este botón sea lo primero que vea el usuario al ingresar a la página.
5. *Include Your Contact Information*: Incluye tu información de contacto. “..., make sure to include clear and easy-to-find contact information on your website”. Una de las principales razones por las que usuarios visitan páginas web es para obtener información sobre una empresa. Por lo anterior es necesario incluir en la página datos como número de teléfono, dirección y emails de contacto. Además de lo anterior, se puede agregar un formulario de contacto que el usuario deba rellenar con su consulta.
6. *Avoid Distracting Autoplay*: Evitar distracciones automáticas. “Instead of enabling autoplay features, simply allow users to click the play button”. Reproducir automáticamente audio o video añade vida y carisma a una página web, pero puede resultar molesto para algunos usuarios, por lo que se recomienda evitar su uso y simplemente dar la opción de reproducción mediante un botón. Además de lo anterior, dificulta la navegación de la página y puede provocar que el usuario pierda el interés y no encuentre el CTA.
7. *Proofread, Proofread, Proofread*: Corregir, Corregir, Corregir. “While the visuals on a site are crucial to its overall design, the text is equally (if not more) important”. Mientras que la parte visual de una página web es crucial para el diseño en general, el texto es igualmente importante (si no más). Errores de ortografía pueden provocar que el usuario pierda interés en el contenido de la página web, además de afectar la credibilidad de la empresa. Para evitar lo mencionado se recomienda pedir una revisión de algún externo y corregir errores encontrados cuanto antes.

Una lectura inicial de estas convenciones permite dar con la idea principal que se intenta aplicar mediante su uso, desarrollar una página web enfocada en el usuario y en lo que se espera o se quiere que éste haga en el sitio.

Un segundo set de convenciones es el presentado por Erik Devaney en (Devaney, 2017). Esta vez, se tiene 8 convenciones sobre el diseño de páginas Web, las cuales están centradas en usabilidad y experiencia del usuario:

1. *Simplicity* (Simplicidad): Menciona que desde un punto de vista de la usabilidad y UX, la simplicidad es recomendable. Tanto desde el punto de vista de los colores como la tipografía y gráficos que se deseen utilizar. Mantener el sitio web simple y elegante, ayuda a que el usuario no se distraiga con información innecesaria y logre su objetivo.
2. *Visual Hierarchy* (Jerarquía Visual): El objetivo de optimizar para usabilidad y UX es que el usuario complete una acción. Junto a esto se debe organizar la página de modo que el usuario navegue de forma natural hacia donde se quiere, siendo el foco de su atención el botón o formulario que debe presionar o completar para lograr el objetivo. Lo anterior se logra ajustando la posición, el color y el tamaño de ciertos elementos, guiando la atención del usuario hacia los CTA (*Call-to-action*).
3. *Navigability* (Navegabilidad): Un estilo de navegación intuitivo del sitio es clave para asegurar que los usuarios encuentren lo que están buscando. Lo ideal es que el usuario no necesite pensar extensivamente sobre su siguiente acción. Para lo anterior plantea una serie de tips:
 - a. Mantener la estructura de navegación principal (menú) simple y cerca de la parte superior de la página.
 - b. Incluir navegación en el footer.
 - c. Utilizar *breadcrumbs* (migajas de pan), para asegurarse que el usuario sepa dónde está y de donde vino en todo momento.
 - d. Incluir un cuadro de búsqueda.
 - e. No ofrecer muchas opciones de navegación, mantenerlo simple.
 - f. No navegar muy profundo, se recomiendan máximo tres niveles de profundidad.
 - g. Incluir links y explicar a donde llevan.
4. *Consistency* (Consistencia): Además de mantener la navegación consistente, es necesario mantener el *look and feel* de la página consistente. Fondos, colores, tipografías son áreas en las que mantener consistencia puede tener un impacto positivo en usabilidad y UX. Se refiere a que cada tipo de página (inicio, informativas, tiendas) debe tener un formato similar, para así permitir que el usuario entienda que tipo de información puede encontrar en la página.
5. *Accessibility* (Accesibilidad): Se debe lograr que la página pueda ser visitada desde múltiples navegadores, sistemas operativos y dispositivos. Lo ideal, es que el desarrollo de la página web sea con un diseño responsive, capaz de adaptarse a cualquier navegador y pantalla adecuando tamaños y posiciones de los distintos componentes. Otra forma de lograr esto es, por ejemplo, añadir texto alternativo

en imágenes en caso de que algún usuario no pueda verlas, y así pueda entender que se quería mostrar.

6. *Conventionality* (Convencionalidad): Se refiere a un par de convenciones específicas que se han ido adoptando con el tiempo:
 - a. Tener la opción principal de navegación en la parte superior o izquierda de la página.
 - b. Tener un logo en la esquina superior izquierda o al centro de la página.
 - c. Tener que ese logo sea presionable y que redirija a la página de inicio.
 - d. Tener links que cambien de color/apariencia cuando se pasa el ratón encima.

Puede ser tentador no considerar ninguna de estas convenciones para intentar ser original y novedoso, pero esto sería probablemente un error. Hay que aprovechar que se tiene información sobre las páginas que las personas suelen visitar y usar este conocimiento para desarrollar páginas similares. Esto a su vez, facilita la navegación, dado que el usuario se siente a gusto y comprende el funcionamiento de la página sin necesitar investigar o pensar mucho.

7. *Credibility* (Credibilidad): Que tan bien catalogada esta la página web, depende de cómo la perciba el usuario y como la califique, respecto a las otras páginas que conoce. El uso de convenciones como las mencionadas en este artículo, pretenden lograr que el usuario visite un tipo de página que le es familiar, logrando de esta forma otorgarle credibilidad a la página. Otro punto a tener en consideración es ser claro y honesto en lo que se está ofreciendo. La clave está en ser directo con el objetivo de la página y dedicarse luego a explicar porque tiene valor lo que se ofrece. Al igual que lo anterior, tener una página con precios hace parecer que el negocio es confiable y legítimo.
8. *User-Centricity* (Centrado en el usuario): Al final del día, usabilidad y UX dependen de las preferencias de los usuarios. Luego, es lógico pensar que el usuario debe ser el centro de atención en el proceso de desarrollo. La realización de pruebas por parte de usuarios reales es clave en este proceso, usando su retroalimentación para modificar y mejorar el sitio.

A modo de idea para finalizar el artículo, Erik recomienda que el desarrollador se preocupe del usuario, teniéndolo en mente durante todo el proceso de desarrollo, poniéndose en sus zapatos y considerando cada decisión de diseño desde el punto de vista del usuario.

Como otro ejemplo de un set de convenciones vistas desde un ámbito internacional, tenemos las presentadas por Marianne en ("Marianne", 2018). Al igual que las anteriores, se tiene un artículo en ingles por lo que se traducirá el título de la convención y se procederá a explicar el punto.

1. *Website Purpose* (Propósito del Sitio): El sitio debe acomodar las necesidades del usuario. Tener una intención clara en cada página ayudará al usuario a interactuar

con lo que se está ofreciendo. Una página puede tener muchos propósitos, pero hay algunos principales:

- a. Describir Experiencia
 - b. Construir Reputación
 - c. Construir Guías
 - d. Ventas y postventa
2. *Simplicity* (Simplicidad): Se refiere a que es primordial si se tiene en mente usabilidad y UX. Hay varias formas lograr simplicidad a través del diseño con 3 aspectos de este:
- a. Color: El color comunica mensajes y produce respuestas emocionales. Encuentra un set de colores que se adecúe a tu marca, con máximo 5 colores. Combinaciones agradables de colores incrementan el compromiso del cliente y lo hacen sentir a gusto.
 - b. Tipografía: Funciona como la interpretación visual de la voz de la marca. Debe ser legible y no usar más de 3 fuentes de letras distintas.
 - c. Imagenaría: Se incluyen fotografías, ilustraciones, videos y todo tipo de gráficos. Todo debería ser expresivo y capturar el espíritu de la compañía, actuando como la personificación de la personalidad de la marca. La mayor cantidad de información de una página se presenta de manera visual y es importante que para una buena primera impresión se use imágenes de alta calidad, en conjunto con dar credibilidad a la página y mostrar profesionalismo.
3. *Navigation* (Navegación): La navegación es clave para mantener visitantes. Si es confusa el usuario se rendirá y buscará lo que necesita en otra página. Mantener la navegación simple, intuitiva y consistente en cada página es clave.
4. *F-Shaped Pattern Reading* (Patrón de Lectura en F): Un patrón de lectura basado en forma de F es la forma más común en que el usuario escanea texto de una página web. Estudios de seguimiento ocular han encontrado que la mayoría de la información que ve el usuario se encuentra en áreas superiores y a la izquierda de la pantalla. La forma de F representa la forma de leer en Occidente, de izquierda a derecha y de arriba hacia abajo. Una página diseñada efectivamente hará que el usuario use su forma natural de leer para lograr que se concentre en lo que se desea.
5. *Visual Hierarchy* (Jerarquía Visual): El ordenamiento de elementos en orden de importancia. Lo anterior se puede lograr con tamaño, color, imágenes, contraste, tipografía, espacios en blanco, textura o estilo. Una de las principales funciones de aplicar jerarquía visual es establecer un punto focal, dejándole en claro al usuario donde debe centrar su atención.

6. *Content* (Contenido): Un diseño web efectivo tiene junto a un buen diseño, buen contenido. El uso de lenguaje convincente puede atraer e influenciar al usuario a convertirse en clientes.
7. *Grid Based Layout* (Diseño basado en Cuadrículas): Espacios cuadriculados ayudan a estructurar el diseño y mantener el contenido organizado alineando elementos en la página logrando un diseño limpio y ordenado. Este tipo de diseño ordena el contenido en una estructura rígida con columnas, secciones que se alinean y se sienten balanceadas, logrando imponer un orden resultando en una página agradable estéticamente.
8. *Load Time* (Tiempo de Carga): Esperar que una página cargue afectará el número de visitas. Según la autora, cerca de la mitad de los potenciales usuarios espera que una página cargue en 2 segundos o menos, potencialmente dejando una página que no cargó en 3 segundos. Como consejo añade que optimizar el tamaño de imágenes ayudará a la página a cargar más rápido.
9. *Mobile Friendly* (“Amigable con celulares”): Traducción literal con ánimos de mantener el análisis del escrito homogéneo. A lo que se refiere la autora es que la mayoría de los potenciales usuarios usan teléfonos o otros dispositivos para navegar la web. Es importante considerar un modelo responsive al momento de construir el sitio web, para asegurarse que el sitio se ajuste a la pantalla en el dispositivo que sea.

Nuevamente se identifican varias convenciones ya vistas en otros artículos evaluados, como lo son la jerarquía visual, el contenido o el diseño responsive. Damos cuenta a esta altura de la investigación aspectos que son importantes para varios autores distintos, lo que indica de alguna manera un estilo de consenso en estos aspectos del diseño de páginas web.

Uno de los conceptos en el que todos los autores consultados concuerdan, tanto a nivel de mencionarlo textualmente o que esté contenido en algún otro de sus conceptos, se encuentra el de Jerarquía Visual. Todos los autores consultados concuerdan en que una correcta disposición del contenido de la página web permitirá al usuario tener la mejor experiencia posible. Este concepto abarca posicionamiento de imágenes, propuestas de valor, navegación, barras de búsqueda, textos y CTAs. El diseño de la página web debe estar centrado en lograr que el usuario lleve a cabo el objetivo con el que accedió a ella, sea este la compra de algún producto, la consulta de alguna información o la obtención de datos de contacto, lográndose esto con una correcta disposición de los distintos componentes de la página.

Por otro lado, tenemos conceptos como Navegabilidad, Simplicidad y Legibilidad, los que están estrechamente ligados al concepto anterior. Estos 3 son mencionados por al menos dos de los autores citados en el escrito. Navegabilidad, como su nombre lo sugiere, abarca detalles claves de la página, como lo son la posición de la herramienta principal de navegación (menú principal), o bien la disposición de un sistema de migajas (*breadcrumbs*)

que sirvan de indicador para el usuario en términos de indicarle en todo momento donde está y de donde vino. Los otros dos conceptos de simplicidad y legibilidad van de la mano y abarcan desde el color y tamaño del texto a la cantidad general de información por página y donde se sitúa. Estos 3 conceptos se enfocan en lograr que la página sea lo más amigable posible en términos de usabilidad y, a su vez, dispongan de la información justa y necesaria de la mejor manera.

Junto a los conceptos anteriores tenemos algunos más enfocados en el funcionamiento del sitio web, como lo son el diseño responsive, tiempos de carga o el *autoplay*. En primer lugar, se podría decir que de una forma u otra cada autor abarca el diseño responsive de distinta forma, pero para todos está claro que es necesario aplicarlo, sobre todo en el mercado de hoy. En una época en que la mayoría de las consultas a páginas web se realizan desde dispositivos móviles, es necesario asegurar que la página web pueda consultarse desde cualquier pantalla y cualquier dispositivo. Para páginas nuevas se reduce simplemente a una decisión de implementación al momento de comenzar el desarrollo. Para páginas ya establecidas puede resultar más complicado, dado que adaptar el contenido ya existente a un modelo responsive puede tomar más trabajo del que tomaría realizarlo de cero.

Cuando se menciona tiempos de carga, se abarca también conceptos como el *autoplay*, la cantidad de imágenes o bien la cantidad de contenido en general. Se debe buscar un balance entre lo que se quiere mostrar y el tiempo que demora esto en cargar en distintos escenarios. Primero que nada, se recomienda no tener elementos audiovisuales que se reproduzcan automáticamente al ingresar a la página, dado que aparte de incrementar los tiempos de carga, distraen al usuario llegando a incomodarlo, provocando incluso una potencial salida anticipada del sitio. Aparte de elementos audiovisuales puede resultar perjudicial tener una página con muchas imágenes o gráficos, dado que, junto con afectar a los tiempos de carga, pueden llegar a sobrecargar el sitio y evitar que el usuario llegue a obtener la información que desea y deje el sitio sin una experiencia satisfactoria.

En términos generales, los 3 conceptos anteriores tratan de ayudar a comprender a un diseñador el cómo un balance adecuado de elementos, junto a su correcta modificación en términos de lugar y tamaño para todo tipo de pantallas ayuda a que el usuario tenga una buena experiencia dentro del sitio.

Pasando a aspectos más generales dentro del diseño de páginas web, se mencionan conceptos como Convencionalidad, Consistencia y Credibilidad. Los 2 primeros corresponden nuevamente a aspectos de una página web que van de la mano. El primero pretende llevar al desarrollador a crear una página que “siga” los estándares ya usados, permitiendo un cierto margen de creatividad, pero manteniendo formatos conocidos. Lo anterior es deseable si se toma en cuenta que el usuario busca una experiencia fácil y rápida. La mejor manera de lograr esto es creando una página similar al resto con un sello distintivo que le permita mantenerse competitiva en el mercado. Consistencia se refiere simplemente a mantener la página de una sola línea, para que al navegar el usuario sienta que sigue en el mismo portal y no ha cambiado. Aspectos como el tipo de letra y los colores utilizados

juegan un rol fundamental, mientras que el diseño general de la página como la distribución de cuadros de texto y posicionamiento de imágenes y CTAs ayudan a cumplir el objetivo.

El concepto de credibilidad hace alusión a la intención que debe tener un desarrollador de entregar información real y verídica, manteniendo al usuario siempre informado sobre lo que se está ofreciendo. Por otro lado, este último concepto también se usa para entender la siguiente idea: Si el usuario encuentra la página similar a otras que ya conoce, le generará más confianza que una página innovadora y desconocida. Lógicamente un usuario deseará que una página desconocida sea navegable de manera similar a otras que maneja, dado que esto le facilitará encontrar lo que necesite y cumplir el objetivo por el cual ingresó a la página en primer lugar.

2 IMPLEMENTACIÓN

2.1 Analizador de páginas web

```
import webbrowser
import pyodbc
import web
import sys
import time
import tkinter as tk
import statistics
from statistics import mode
from tkinter import ttk
from tkinter import messagebox
from datetime import datetime
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.color import Color
from selenium.webdriver.chrome.options import Options
from selenium.common.exceptions import NoSuchElementException
from selenium.common.exceptions import WebDriverException
from selenium.common.exceptions import TimeoutException
from selenium.common.exceptions import InvalidArgumentException
from colormath.color_objects import sRGBColor, LabColor
from colormath.color_conversions import convert_color
from colormath.color_diff import delta_e_cie2000

elementos = ["header", "footer", "logo", "search", "background",
"letra", "font_size"]

#inicializar parametros webdriver
url = ""
options = Options()
#options.add_argument("--headless")
driver = webdriver.Chrome(options=options)
driver.set_page_load_timeout(60)

#configuracion conexión base de datos
conexion = pyodbc.connect('Driver={SQL Server};'
    'Server=DESKTOP-VNN7TSG\\SQLEXPRESS;'
    'Database=Test;'
    'Trusted_Connection=yes;')
cursor = conexion.cursor()

#variables globales
header_param = ""
footer_param = ""
search_param = ""
logo_param = ""
loadtime_param = ""
background_color_param = ""
letra_param = ""
font_size_param = ""
```

```
def main():
    global url
    url = mystring.get()
    get_pagina(url)
    cursor.execute("DELETE FROM Test.dbo.resultados")
    print("URL a analizar: %s" %url)
    add_result("url", url)
    time.sleep(4)
    print("\nObteniendo parametros...")
    get_parametros()
    print("\nSe inicia el proceso de análisis...")
    get_loadtime()
    check_infinite()
    get_window_width()
    get_header(header_param)
    get_footer(footer_param)
    get_searchbox(search_param)
    get_logo(logo_param)
    get_background_color(background_color_param)
    get_color_letra(letra_param)
    get_font_size(font_size_param)
    get_font_family()
    get_link()
    get_video()
    get_audio()
    get_publicidad(url)
    get_text_adjust()
    get_h1()
    webbrowser.open("http://localhost:8080/", new=1)
    cerrar()

def get_text_adjust():
    print("Buscando ajuste de texto")
    reglas = ["-webkit-text-size-adjust", "-ms-text-size-adjust", "-moz-text-size-adjust", "text-size-adjust"]
    exitos = []
    fallas = []
    try:
        body = driver.find_element_by_tag_name("body")
        properties = driver.execute_script('return window.getComputedStyle(arguments[0], null);', body)
        for propiedad in properties:
            if propiedad in reglas and
body.value_of_css_property(propiedad) == 'none':
                fallas.append(propiedad+" "+body.value_of_css_property(propiedad))
                print("Se encontró una regla MALA: %s: %s"
%(propiedad, body.value_of_css_property(propiedad)))
            elif propiedad in reglas and
body.value_of_css_property(propiedad) == '100%':
                exitos.append(propiedad+" "+body.value_of_css_property(propiedad))
                print("Se encontró una regla BUENA: %s: %s"
%(propiedad, body.value_of_css_property(propiedad)))
```

```
print("Exitos: %s"%exitos)
add_result("text_adjust_bueno", ("", ".join(exitos)))
if len(exitos) == 0:
    add_comentario("text_adjust_bueno", "No se encontró
propiedades que fijen bien el tamaño de la letra")
elif len(exitos) == 1:
    add_comentario("text_adjust_bueno", "Se encontró 1
propiedad que fija bien el tamaño de la letra: %s" %("",
".join(exitos)))
elif len(exitos) >= 1:
    add_comentario("text_adjust_bueno", "Se encontró %s
propiedades que fijan bien el tamaño de la letra: %s" %(len(exitos),
(",", ".join(exitos))))
print("Fallas: %s"%fallas)
add_result("text_adjust_malo", ("", ".join(fallas)))
if len(fallas) == 0:
    add_comentario("text_adjust_malo", "No se encontró
propiedades que fijen mal el tamaño de la letra")
elif len(fallas) == 1:
    add_comentario("text_adjust_bueno", "Se encontró 1
propiedad que fija mal el tamaño de la letra: %s" %("",
".join(fallas)))
elif len(fallas) >= 1:
    add_comentario("text_adjust_bueno", "Se encontró %s
propiedad(es) que fijan mal el tamaño de la letra: %s" %(len(fallas),
(",", ".join(fallas))))

except NoSuchElementException:
    print("No se encontró text_adjust")
    return

def get_video():
    print("Buscando videos")
    vid = 0
    try:
        videos = driver.find_elements_by_tag_name("video")
        for video in videos:
            if video.get_attribute("autoplay"):
                vid += 1
        print("Videos Automáticos: %s"%vid)
        add_result("video", vid)
        if vid == 0:
            add_comentario("video", "No se detectó videos que se
reprodujeran automáticamente")
        elif vid > 0 :
            add_comentario("video", "Se encontró %s video(s) con
reproducción automática" %vid)
    except NoSuchElementException:
        add_result("video", vid)
        add_comentario("video", "No se encontró videos en la
página")
    print("No se encontró videos")
    return
```

```
def get_audio():
    print("Buscando Audios")
    aud = 0
    try:
        audios = driver.find_elements_by_tag_name("audio")
        for audio in audios:
            if audio.get_attribute("autoplay"):
                aud += 1
        print("Audios Automáticos: %s"%aud)
        add_result("audio", aud)
        if aud == 0:
            add_comentario("audio", "No se detectó pistas de
audio que se reprodujeran automaticamente")
        elif aud > 0 :
            add_comentario("audio", "Se encontró %s pista(s) de
audio(s) con reproducción automática" %aud)
        except NoSuchElementException:
            add_result("audio", aud)
            add_comentario("audio", "No se encontró pistas de audio en
la página")
        print("No se encontró pistas de audio en la página")
        return

def get_font_family():
    try:
        font_family =
driver.find_element_by_tag_name("body").value_of_css_property("font-
family")
        add_result("font_family", font_family)
        if "SANS-SERIF" in font_family.upper():
            add_comentario("font_family", "Se encontró la fuente
de texto esperada dentro de la familia de fuentes")
        else:
            add_comentario("font_family", "La familia de fuentes
encontrada no contiene la fuente esperada")
            print("font_family encontrado: %s / Esperado: Sans-serif"
%font_family)
        except NoSuchElementException:
            print("No se encontró font_family")
            return

def parse_rgba(rgba):
    sin_rgba = rgba[5:-4]
    split = sin_rgba.split(",")
    return split

def get_link():
    print("Obteniendo links")
    global color
    split_letra = parse_rgba(color)
    x = float(split_letra[0])
    y = float(split_letra[1])
    z = float(split_letra[2])
    rgb_letra_parseado = sRGBColor(x, y, z, is_upscaled=True)
```

```

lab2 = convert_color(rgb_letra_parseado, LabColor)
rgb_azul = sRGBColor(0, 0, 255, is_upscaled=True)
lab_azul = convert_color(rgb_azul, LabColor)
nr_links = 0
cumple = 0
no_cumple = 0
cumple_color = 0
cumple_sub = 0
suf_color = 0
try:
    links = driver.find_elements_by_xpath("//a[@href]")
    for link in links:
        if link.text != "":
            nr_links += 1
            color_link =
link.value_of_css_property("color")
            split = parse_rgba(color_link)
            a = float(split[0])
            b = float(split[1])
            c = float(split[2])
            rgb_parseado = sRGBColor(a, b, c,
is_upscaled=True)
            lab1 = convert_color(rgb_parseado, LabColor)
            delta_e = delta_e_cie2000(lab1, lab_azul)
            if delta_e <= 10:
                cumple_color += 1
                if link.value_of_css_property("text-
decoration") == "underline":
                    cumple_sub += 1
                    cumple += 1
                if link.value_of_css_property("text-
decoration") != "underline":
                    no_cumple += 1
            if delta_e > 10:
                no_cumple += 1
                new_delta_e = delta_e_cie2000(lab1, lab2)
                if new_delta_e > 10:
                    suf_color += 1
                if link.value_of_css_property("text-
decoration") == "underline":
                    cumple_sub += 1
            print("Nr links: ", nr_links)
            add_result("nr_links", nr_links)
            add_comentario("nr_links", "La página posee %s links"
%nr_links)
            print("Cantidad de links que cumplen con todo: ", cumple)
            add_result("cumple", cumple)
            if cumple == nr_links:
                add_comentario("cumple", "Todos los links cumplen con
los esperado")
            if no_cumple == nr_links:
                add_comentario("cumple", "Ningún link cumple con lo
esperado")
            if cumple < nr_links and no_cumple < nr_links:

```

```
        add_comentario("cumple", "Se encontraron %s links que
cumplen con lo esperado y %s links que no" %(cumple, no_cumple))
        print("Cantidad de links que no cumplen: ", no_cumple)
        add_result("no_cumple", no_cumple)
        print("Links subrayados: ", cumple_sub)
        add_result("cumple_sub", cumple_sub)
        if cumple_sub == nr_links:
            add_comentario("cumple_sub", "Todos los links están
subrayados")
        if cumple_sub != nr_links:
            add_comentario("cumple_sub", "El numero de links
subrayados es: %s" %cumple_sub)
            print("Links con color correcto: ", cumple_color)
            add_result("cumple_color", cumple_color)
            if cumple_color == nr_links:
                add_comentario("cumple_color", "Todos los links son
de color azul")
            if cumple_color != nr_links:
                add_comentario("cumple_color", "Los links que cumplen
con el color azul son: %s" %cumple_color)
                print("Links con color suficientemente diferente: %s \n"
%suf_color)
                add_result("suf_color", suf_color)
                add_comentario("suf_color", "Los links que tienen un color
distinto al azul pero suficientemente distinto al color de la letra
son: %s" %suf_color)

    except NoSuchElementException:
        print("No se encontró ningún link \n")
        return

def get_publicidad(url):
    print("Buscando Publicidad")
    largo = len(url)
    publicidad = 0
    try:
        links = driver.find_elements_by_tag_name("iframe")
        for link in links:
            if link.value_of_css_property("display") != "none":
                publicidad += 1
        print("Publicidad encontrada: %s"%publicidad)
        add_result("publicidad", publicidad)
        if publicidad == 0:
            add_comentario("publicidad", "No se encontró anuncios
en la página")
        if publicidad > 0:
            add_comentario("publicidad", "Se encontró %s
anuncio(s) en la página" %publicidad)

    except NoSuchElementException:
        add_result("publicidad", publicidad)
        add_comentario("publicidad", "No se encontró anuncios en
la página")
        print("No se encontró publicidad")
```

```

        return

def get_window_width():
    ancho = driver.execute_script("return
document.body.clientWidth")
    ancho_ventana = driver.get_window_size()["width"]
    ancho_ventana_recomendado = int((ancho_ventana)*0.8)
    print("Ancho ventana recomendado: %s"
%ancho_ventana_recomendado)
    add_result("window_width", f"{ancho:,}".replace(',',' '.'))
    add_comentario("window_width", "El ancho de la página es %s, el
ancho de la pantalla es %s y el ancho recomendado de la página es
%s"%(f"{ancho:,}".replace(',',' '.'),f"{ancho_ventana:,}".replace(',','
.'), f"{ancho_ventana_recomendado:,}".replace(',',' '.')))

def check_infinite():
    driver.maximize_window()
    ventana = driver.get_window_size()["height"]
    scrolls = 0
    last_height = driver.execute_script("return
document.body.scrollHeight")
    if last_height > ventana*3:
        add_result("window_height",
f"{last_height:,}".replace(',',' '.'))
        add_comentario("window_height", "Página mas alta de lo
recomendado. La altura de la página es %s pixeles y La altura
recomendado es de no mas de 3 pantallas completas. La altura de la
pantalla utilizada es de %s pixeles" %(f"{last_height:,}".replace(',','
.'), f"{ventana:,}".replace(',',' '.')))
    if last_height <= ventana*3:
        add_result("window_height",
f"{last_height:,}".replace(',',' '.'))
        add_comentario("window_height", "La altura de la página se
encuentra dentro de lo recomendado: %s pixeles"
%f"{ventana*3:,}".replace(',',' '.'))
        while True:
            driver.execute_script("window.scrollTo(0,
document.body.scrollHeight);")
            scrolls += 1
            time.sleep(2)
            new_height = driver.execute_script("return
document.body.scrollHeight")
            if new_height == last_height:
                print("Se llegó al final en :%s scroll(s)" %scrolls)
                add_result("infinite", "No")
                add_comentario("infinite", "Se llegó al final de la
página en %s scroll(s) por lo que no tiene infinite scrolling"
%scrolls)
                break
            last_height = new_height
        if scrolls > 2:
            print("La pagina tiene infinite scrolling")
            add_result("infinite", "Si")

```

```

        add_comentario("infinite", "La página posee infinite
scrolling: %s scrolls" %scrolls)
        break

def evaluar_logo(ancho, alto):
    area = int(ancho * alto)
    if area > 14400:
        return("Logo mas grande de lo esperado: %s > 14.400"
%f"{area:,}".replace(',','.'))
    if area < 14400:
        return("Logo mas pequeño de lo esperado: %s < 14.400"
%f"{area:,}".replace(',','.'))
    if area == 14400:
        return("El logo tiene el tamaño perfecto: %s"
%f"{area:,}".replace(',','.'))

def add_comentario(item, comentario):
    cursor.execute('UPDATE Test.dbo.resultados SET comentario = ?
WHERE item = ? ', comentario, item)

def add_result(item, resultado):
    cursor.execute("INSERT INTO Test.dbo.resultados (item,
resultado) VALUES (?,?)", item, resultado)

def add dimensiones(item, elemento):
    altura = item.rect["height"]
    ancho = item.rect["width"]
    ubicacion_y = item.location["y"]
    ubicacion_x = item.location["x"]
    cursor.execute("INSERT INTO Test.dbo.resultados (item,
resultado) VALUES (?,?)", elemento+"_height",
f"{altura:,}".replace(',','.'))
    cursor.execute("INSERT INTO Test.dbo.resultados (item,
resultado) VALUES (?,?)", elemento+"_width", f"{ancho:,}".replace(',','.'))
    cursor.execute("INSERT INTO Test.dbo.resultados (item,
resultado) VALUES (?,?)", elemento+"_x",
f"{ubicacion_x:,}".replace(',','.'))
    cursor.execute("INSERT INTO Test.dbo.resultados (item,
resultado) VALUES (?,?)", elemento+"_y",
f"{ubicacion_y:,}".replace(',','.'))
    if elemento == "header":
        if float(ubicacion_x) == 0 and float(ubicacion_y) == 0:
            add_comentario(elemento+"_x", "El %s se encuentra en
la posición esperada: (%s, %s) == (0,0)" %(elemento, ubicacion_x,
ubicacion_y))
        if float(ubicacion_x) != 0 or float(ubicacion_y) != 0:
            add_comentario(elemento+"_x", "El %s no se encuentra
en la posición esperada: (%s, %s) == (0,0)" %(elemento, ubicacion_x,
ubicacion_y))
        area = int(altura * ancho)
        area_esperada =
int(int(header_width_param.replace('.', '')) *
int(header_height_param))

```

```

print(area_esperada)
if area < area_esperada:
    add_comentario("header_height", "El header es mas
pequeño de lo esperado: %s < %s"%(f"{area:,}".replace(',','.'),
f"{area_esperada:,}".replace(',','.')))
    if area > area_esperada:
        add_comentario("header_height", "El header es mas
grande de lo esperado: %s > %s"%(f"{area:,}".replace(',','.'),
f"{area_esperada:,}".replace(',','.')))
    if elemento == "footer":
        height = driver.execute_script("return
document.body.scrollHeight")
        y_esperado = (float(height) - float(altura))
        if float(ubicacion_x) == 0 and float(ubicacion_y) ==
y_esperado:
            add_comentario(elemento+"_x", "El %s se encuentra en
la posición esperada dado que la pagina mide %s pixeles de altura"
%(elemento, f"{height:,}".replace(',','.')))
            if float(ubicacion_x) != 0 or float(ubicacion_y) !=
(driver.execute_script("return document.body.scrollHeight") -
float(altura)):
                add_comentario(elemento+"_x", "El %s no se encuentra
en la posición esperada, dado que se espera que su coordenada en 'x'
sea %s y su coordenada en 'y' sea la resta entre la altura de la
página de %s pixeles y su altura de %s pixeles: %s" %(elemento,
f"{int(footer_x_media_param):,}".replace(',','.'),
f"{int(height):,}".replace(',','.'), altura,
f"{int(y_esperado):,}".replace(',','.')))
                area = int(altura * ancho)
                area_esperada =
int(int(footer_width_param.replace('.', '')) *
int(footer_height_param.replace('.', '')))
                print(area_esperada)
                if area < area_esperada:
                    add_comentario("footer_height", "El footer es mas
pequeño de lo esperado: %s < %s"%(f"{area:,}".replace(',','.'),
f"{area_esperada:,}".replace(',','.')))
                    if area > area_esperada:
                        add_comentario("footer_height", "El footer es mas
grande de lo esperado: %s > %s"%(f"{area:,}".replace(',','.'),
f"{area_esperada:,}".replace(',','.')))
                    if elemento == "logo":
                        comentario =
evaluar_logo(float(item.rect["width"]),float(item.rect["height"]))
                        add_comentario(elemento+"_height", comentario)
                        if float(ubicacion_x) == 0 and float(ubicacion_y) == 0:
                            add_comentario(elemento+"_x", "El %s se encuentra en
la posición esperada (Superior izquierda)" %elemento)
                            add_comentario(elemento+"_y", "El %s se encuentra en
la posición esperada (Superior izquierda)" %elemento)
                        if float(ubicacion_x) != 0 or float(ubicacion_y) != 0:
                            add_comentario(elemento+"_x", "El %s no se encuentra
en la posición esperada (Superior izquierda)" %elemento)

```

```
        add_comentario(elemento+"_y", "El %s no se encuentra
en la posición esperada (Superior izquierda)" %elemento)
```

```
def cerrar():
```

```
    conexion.commit()
    driver.close()
    driver.quit()
    win.destroy()
```

```
def get_pagina(input):
```

```
    try:
        driver.get(input)
    except InvalidArgumentException:
        messagebox.showwarning("Warning","URL inválida")
        driver.close()
        driver.quit()
        sys.exit()
    except WebDriverException:
        messagebox.showwarning("Warning","No se pudo acceder a la
pagina")
        driver.close()
        driver.quit()
        sys.exit()
```

```
def get_parametros():
```

```
    global header_param, footer_param, search_param, logo_param,
loadtime_param, background_color_param, letra_param, font_size_param,
page_height_param, page_width_param, search_height_param,
header_height_param, header_width_param, header_x_param,
header_y_param, footer_height_param, footer_width_param,
footer_x_param, footer_y_param, search_width_param, search_x_param,
search_y_param, logo_height_param, logo_width_param, logo_x_param,
logo_y_param, header_x_media_param, header_y_media_param,
footer_x_media_param, footer_y_media_param, search_x_media_param,
search_y_media_param, logo_x_media_param, logo_y_media_param
    cursor.execute('SELECT * FROM Test.dbo.parametros')
    for row in cursor:
        if row[0] == "header":
            header_param = row[1]
        if row[0] == "footer":
            footer_param = row[1]
        if row[0] == "searchbox":
            search_param = row[1]
        if row[0] == "logo":
            logo_param = row[1]
        if row[0] == "loadtime":
            loadtime_param = row[1]
        if row[0] == "background":
            background_color_param = row[1]
        if row[0] == "letra":
            letra_param = row[1]
        if row[0] == "font_size":
            font_size_param = row[1]
        if row[0] == "page_height":
```

```
        page_height_param = row[1]
if row[0] == "page_width":
    page_width_param = row[1]
if row[0] == "search_height":
    search_height_param = row[1]
if row[0] == "header_height":
    header_height_param = row[1]
if row[0] == "header_width":
    header_width_param = row[1]
if row[0] == "header_x":
    header_x_param = row[1]
if row[0] == "header_y":
    header_y_param = row[1]
if row[0] == "footer_height":
    footer_height_param = row[1]
if row[0] == "footer_width":
    footer_width_param = row[1]
if row[0] == "footer_x":
    footer_x_param = row[1]
if row[0] == "footer_y":
    footer_y_param = row[1]
if row[0] == "search_width":
    search_width_param = row[1]
if row[0] == "search_x":
    search_x_param = row[1]
if row[0] == "search_y":
    search_y_param = row[1]
if row[0] == "logo_height":
    logo_height_param = row[1]
if row[0] == "logo_width":
    logo_width_param = row[1]
if row[0] == "logo_x":
    logo_x_param = row[1]
if row[0] == "logo_y":
    logo_y_param = row[1]
if row[0] == "header_x_media":
    header_x_media_param = row[1]
if row[0] == "header_y_media":
    header_y_media_param = row[1]
if row[0] == "footer_x_media":
    footer_x_media_param = row[1]
if row[0] == "footer_y_media":
    footer_y_media_param = row[1]
if row[0] == "search_x_media":
    search_x_media_param = row[1]
if row[0] == "search_y_media":
    search_y_media_param = row[1]
if row[0] == "logo_x_media":
    logo_x_media_param = row[1]
if row[0] == "logo_y_media":
    logo_y_media_param = row[1]
print("Header: %s" %header_param)
print("Footer: %s" %footer_param)
print("Searchbox: %s" %search_param)
```

```
print("Logo: %s" %logo_param)
print("Loadtime: %s" %loadtime_param)
print("Background Color: %s" %background_color_param)
print("Letra: %s" %letra_param)
print("Font Size: %s" %font_size_param)

def get_header(parametro):
    if parametro == "tag": buscar_tag("header")
    if parametro == "id": buscar_id("header")
    if parametro == "class": buscar_class("header")

def get_footer(parametro):
    if parametro == "tag": buscar_tag("footer")
    if parametro == "id": buscar_id("footer")
    if parametro == "class": buscar_class("footer")

def get_searchbox(parametro):
    if parametro == "tag": buscar_tag("search")
    if parametro == "id": buscar_id("search")
    if parametro == "class": buscar_class("search")

def get_logo(parametro):
    if parametro == "tag": buscar_tag("logo")
    if parametro == "id": buscar_id("logo")
    if parametro == "class": buscar_class("logo")

def get_loadtime():
    inicio_carga = driver.execute_script("return
window.performance.timing.responseStart")
    fin_carga = driver.execute_script("return
window.performance.timing.domComplete")
    loadtime = (fin_carga - inicio_carga)/1000
    if (loadtime < 0):
        loadtime = 0
    add_result("loadtime", str(loadtime).replace('.', ','))
    if loadtime <= float(loadtime_param.replace('.', ',')):
        add_comentario("loadtime", "Tiempo de carga menor al
esperado de %s segundos" %str(loadtime_param).replace('.', ','))
    if loadtime > float(loadtime_param.replace('.', ',')):
        add_comentario("loadtime", "Tiempo de carga mayor al
esperado de %s segundos" %str(loadtime_param).replace('.', ','))
    print("Tiempo de carga: %s segundos" %loadtime)

def get_background_color(parametro):
    try:
        background_color =
driver.find_element_by_tag_name("body").value_of_css_property("backgro
und-color")
        add_result("background_color", background_color)
        if background_color != background_color_param:
            add_comentario("background_color", "El color de fondo
no es igual al esperado: %s" %background_color_param)
        if background_color == background_color_param:
```

```
        add_comentario("background_color", "El color de fondo
es igual al esperado: %s" %background_color_param)
        print("Background-color encontrado: %s / Esperado: %s" %
(background_color, parametro))
    except NoSuchElementException:
        print("No se encontró background-color")
        return

def get_color_letra(parametro):
    global color
    try:
        color =
driver.find_element_by_tag_name("body").value_of_css_property("color")
        add_result("color", color)
        if color != letra_param:
            add_comentario("color", "El color de la letra no es
igual al esperado: %s" %letra_param)
            if color == letra_param:
                add_comentario("color", "El color de la letra es
igual al esperado: %s" %letra_param)
                print("Color letra encontrado: %s / Esperado: %s" %
(color, parametro))
    except NoSuchElementException:
        print("No se encontró color")
        return

def get_font_size(parametro):
    try:
        font_size =
driver.find_element_by_tag_name("body").value_of_css_property("font-
size")
        add_result("font_size", font_size)
        if font_size <= font_size_param:
            add_comentario("font_size", "El tamaño de la letra es
menor al esperado: %s" %font_size_param)
            if font_size >= font_size_param:
                add_comentario("font_size", "El tamaño de la letra es
mayor al esperado: %s" %font_size_param)
            if font_size == font_size_param:
                add_comentario("font_size", "El tamaño de la letra es
igual al esperado")
                print("Tamaño letra encontrado: %s / Esperado: %s" %
(font_size, parametro))
    except NoSuchElementException:
        print("No se encontró font-size")
        return

def buscar_tag(elemento):
    print("Buscando %s por tag_name" %elemento)
    try:
        item = driver.find_element_by_tag_name(elemento)
        add_result(elemento, "Si")
        add dimensiones(item, elemento)
```

```
        add_comentario(elemento, "Se encontró correctamente el %s
por el tag_name" %elemento)
        print("Se encontró " + elemento + " en x: " +
str(driver.find_element_by_tag_name(elemento).location["x"]) +
        " y: " +
str(driver.find_element_by_tag_name(elemento).location["y"]))
    except NoSuchElementException:
        add_result(elemento, "No")
        add_comentario(elemento, "No se encontró el %s buscando
por tag_name" %elemento)
        print("No se encontró %s" %elemento)
        return

def buscar_id(elemento):
    print("Buscando %s por id" %elemento)
    try:
        item = driver.find_element_by_id(elemento)
        add_result(elemento, "Si")
        add dimensiones(item, elemento)
        add_comentario(elemento, "Se encontró correctamente el %s
por el id" %elemento)
        print("Se encontró " + elemento + " en x: " +
str(driver.find_element_by_id(elemento).location["x"]) +
        " y: " +
str(driver.find_element_by_id(elemento).location["y"]))
    except NoSuchElementException:
        add_result(elemento, "No")
        add_comentario(elemento, "No se encontró el %s buscando
por id" %elemento)
        print("No se encontró %s" %elemento )
        return

def buscar_class(elemento):
    print("Buscando %s por class_name" %elemento)
    try:
        item = driver.find_element_by_class_name(elemento)
        add_result(elemento, "Si")
        add dimensiones(item, elemento)
        add_comentario(elemento, "Se encontró correctamente el %s
por el class_name" %elemento)
        print("Se encontró " + elemento + " en x: " +
str(driver.find_element_by_class_name(elemento).location["x"]) +
        " y: " +
str(driver.find_element_by_class_name(elemento).location["y"]))
    except NoSuchElementException:
        add_result(elemento, "No")
        add_comentario(elemento, "No se encontró el %s buscando
por class_name" %elemento)
        print("No se encontró %s" %elemento )
        return

def get_h1():
    print("Buscando h1")
    try:
```

```
h1 = driver.find_element_by_tag_name("h1")
texto = h1.text
if texto != "":
    print("Se encontró un h1: %s"%texto)
    add_result("h1", "Sí")
    add_comentario("h1", "Se encontró correctamente al
menos un h1: %s"%texto)
if texto == "":
    print("Se encotró un h1 vacío")
    add_result("h1", "No")
    add_comentario("h1", "El h1 encontrado esta vacío")

except NoSuchElementException:
    print("No se encontró h1")
    add_result("h1", "No")
    add_comentario("h1", "No se encontró h1")
    return

win = tk.Tk()
win.title("Evaluación Heurística de Páginas Web")
window_height = 100
window_width = 450
screen_width = win.winfo_screenwidth()
screen_height = win.winfo_screenheight()
x_cordinate = int((screen_width/2) - (window_width/2))
y_cordinate = int((screen_height/2) - (window_height/2))
win.geometry("{}x{}+{}+{}".format(window_width, window_height,
x_cordinate, y_cordinate))

entry_frame = ttk.LabelFrame(win, text="Ingrese URL", relief=tk.RIDGE)
entry_frame.grid(row=1, column=1, columnspan = 2, pady = 10, padx =
10, sticky=tk.E + tk.W + tk.N + tk.S)
entry_label = ttk.Label(entry_frame, text="URL")
entry_label.grid(row=1, column=1, sticky=tk.E + tk.W + tk.N + tk.S)
mystring = tk.StringVar(win)
my_entry = ttk.Entry(entry_frame, textvariable=mystring, width=65)
my_entry.grid(row=1, column=2, sticky=tk.W, pady=3)
buscar = ttk.Button(win, text="Buscar", command=main)
buscar.grid(row=3, column=1)
salir = ttk.Button(win, text="Salir", command=cerrar)
salir.grid(row=3, column=2)

win.mainloop()
```

2.2 Llenado de datos para confección de parametría

```
import webbrowser
import pyodbc
import time
from datetime import datetime
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.options import Options
from selenium.common.exceptions import NoSuchElementException
from selenium.common.exceptions import WebDriverException
from selenium.common.exceptions import TimeoutException

#configuración webdriver y paginas

urls = ["https://www.lun.com", "https://www.emol.com",
"https://www.latercera.cl",
        "https://www.bancoestado.cl",
"https://www.biobiochile.cl", "https://www.bancochile.cl",
        "https://www.economicos.cl", "https://www.santander.cl",
"https://www.sii.cl",
        "https://www.linio.cl", "https://www.falabella.com",
"https://www.yapo.cl",
        "https://www.gob.cl", "https://www.elmercurio.com",
"https://www.portalinmobiliario.com",
        "https://www.cooperativa.cl", "https://www.nic.cl",
"https://www.lacuarta.com",
        "https://www.transbank.cl", "https://www.paris.cl",
"https://www.guioteca.com",
        "https://www.theclinic.cl", "https://www.ripley.cl",
"https://www.13.cl",
        "https://www.elmostrador.cl", "https://www.eldefinido.cl",
"https://www.24horas.cl",
        "https://www.chileautos.cl", "https://www.uchile.cl",
"https://www.fayerwayer.com",
        "https://www.groupon.cl", "https://www.servipag.com",
"https://www.pcfactory.cl",
        "https://www.uc.cl", "https://www.sodimac.cl",
"https://www.lasegunda.com",
        "https://www.correos.cl", "https://www.soychile.cl",
"https://www.publimetro.cl",
        "https://www.eldinamo.cl", "https://www.elciudadano.cl",
"https://www.glamorama.cl",
        "https://www.chw.net", "https://www.chilevision.cl",
"https://www.tvn.cl",
        "https://www.lanacion.cl", "https://www.df.cl",
"https://www.bancofalabella.com",
        "https://www.cmrfalabella.com", "https://www.lider.cl"]

elementos = ["header", "footer", "logo", "search", "background",
"letra", "font_size"]

options = Options()
#options.add_argument("--headless")
driver = webdriver.Chrome(options=options)
```

```
driver.set_page_load_timeout(60)

#configuracion conexión base de datos

conexion = pyodbc.connect('Driver={SQL Server};'
    'Server=LAPTOP-CH0B4N23\SQLEXPRESS;'
    'Database=Test;'
    'Trusted_Connection=yes;')
cursor = conexion.cursor()
tag_counter = 0
class_counter = 0
id_counter = 0

def main():
    cursor.execute("DELETE FROM Test.dbo.header")
    cursor.execute("DELETE FROM Test.dbo.footer")
    cursor.execute("DELETE FROM Test.dbo.loadtime")
    cursor.execute("DELETE FROM Test.dbo.logo")
    cursor.execute("DELETE FROM Test.dbo.searchbox")
    cursor.execute("DELETE FROM Test.dbo.color")
    cursor.execute("DELETE FROM Test.dbo.font_size")
    cursor.execute("DELETE FROM Test.dbo.dimensiones")
    cursor.execute("DELETE FROM Test.dbo.ubicacion")
    for i, url in enumerate(urls):
        print(i)
        try:
            driver.get(url)
        except WebDriverException:
            print("No se pudo acceder a la página " + url + "\n")
            continue
        inicio_carga = driver.execute_script("return
window.performance.timing.responseStart")
        fin_carga = driver.execute_script("return
window.performance.timing.domComplete")
        loadtime = (fin_carga - inicio_carga)/1000
        if (loadtime < 0):
            loadtime = 0

        print("Tiempo de carga: %s segundos \n" % loadtime)
        cursor.execute("INSERT INTO Test.dbo.loadtime (url,
tiempo) VALUES (?,?)", url, loadtime)

        driver.maximize_window()
        time.sleep(4)
        page_height = driver.execute_script("return
document.body.scrollHeight")
        page_width = driver.execute_script("return
document.body.clientWidth")
        cursor.execute("INSERT INTO Test.dbo.dimensiones (url,
page_height, page_width) VALUES (?, ?, ?)", url, page_height,
page_width)
        cursor.execute("INSERT INTO Test.dbo.ubicacion (url)
VALUES (?)", url)
```

```

        for elemento in elementos:
            global tag_counter, class_counter, id_counter,
background_color, color, font_size
            tag_counter = 0
            class_counter = 0
            id_counter = 0
            background_color = ""
            color = ""
            font_size = ""
            print("Buscando %s en: %s" % (elemento, url))
            if (elemento == "header"):
                header_item = buscar_tag(elemento, url)
                print("tag_counter: " + str(tag_counter) + ",
id_counter: " + str(id_counter) + ", class_counter: " +
str(class_counter) + "\n")
                cursor.execute("INSERT INTO Test.dbo.header
(url, tag, id, class) VALUES (?, ?, ?, ?)", url, tag_counter, id_counter,
class_counter)
            if (elemento == "footer"):
                footer_item = buscar_tag(elemento, url)
                print("tag_counter: " + str(tag_counter) + ",
id_counter: " + str(id_counter) + ", class_counter: " +
str(class_counter) + "\n")
                cursor.execute("INSERT INTO Test.dbo.footer
(url, tag, id, class) VALUES (?, ?, ?, ?)", url, tag_counter, id_counter,
class_counter)
            if (elemento == "logo"):
                log_item = buscar_tag(elemento, url)
                print("tag_counter: " + str(tag_counter) + ",
id_counter: " + str(id_counter) + ", class_counter: " +
str(class_counter) + "\n")
                cursor.execute("INSERT INTO Test.dbo.logo (url,
tag, id, class) VALUES (?, ?, ?, ?)", url, tag_counter, id_counter,
class_counter)
            if (elemento == "search"):
                search_item = buscar_tag(elemento, url)
                print("tag_counter: " + str(tag_counter) + ",
id_counter: " + str(id_counter) + ", class_counter: " +
str(class_counter) + "\n")
                cursor.execute("INSERT INTO Test.dbo.searchbox
(url, tag, id, class) VALUES (?, ?, ?, ?)", url, tag_counter, id_counter,
class_counter)
            if (elemento == "background"):
                buscar_background_color()
                cursor.execute("INSERT INTO Test.dbo.color
(url, background) VALUES (?, ?)", url, background_color)
            if (elemento == "letra"):
                buscar_color()
                cursor.execute("UPDATE Test.dbo.color SET letra
= ? WHERE url = ? ", color, url)
            if (elemento == "font_size"):
                buscar_font_size()
                cursor.execute("INSERT INTO Test.dbo.font_size
(url, font_size) VALUES (?, ?)", url, font_size)

```

```
        print("Finalizó la búsqueda en: " + url + "\n")

def buscar_tag(elemento, url):
    global tag_counter
    try:
        result = driver.find_element_by_tag_name(elemento)
        get dimensiones(result, elemento, url)
        print("Se encontró %s en x: %s, y: %s " % (elemento,
str(result.location["x"]), str(result.location["y"])))
        tag_counter = 1
        return
    except NoSuchElementException:
        print("No se encontró %s en tag name" %elemento)
        buscar_id(elemento, url)

def buscar_id(elemento, url):
    global id_counter
    try:
        result = driver.find_element_by_id(elemento)
        get dimensiones(result, elemento, url)
        print("Se encontró %s en x: %s, y: %s " % (elemento,
str(result.location["x"]), str(result.location["y"])))
        id_counter = 1
        return
    except NoSuchElementException:
        print("No se encontró %s en id" %elemento)
        buscar_class(elemento, url)

def buscar_class(elemento, url):
    global class_counter
    try:
        result = driver.find_element_by_class_name(elemento)
        get dimensiones(result, elemento, url)
        print("Se encontró %s en x: %s, y: %s " % (elemento,
str(result.location["x"]), str(result.location["y"])))
        class_counter = 1
        return
    except NoSuchElementException:
        print("No se encontró %s en class name" %elemento)
        return

def buscar_background_color():
    global background_color
    try:
        background_color =
driver.find_element_by_tag_name("body").value_of_css_property("backgro
und-color")
        print("Background-color: %s" %background_color)
    except NoSuchElementException:
        print("No se encontró background-color")
        return

def buscar_color():
    global color
```

```
try:
    color =
driver.find_element_by_tag_name("body").value_of_css_property("color")
    print("Color letra: %s" %color)
except NoSuchElementException:
    print("No se encontró color")
    return

def buscar_font_size():
    global font_size
    try:
        font_size =
driver.find_element_by_tag_name("body").value_of_css_property("font-
size")
        print("Tamaño letra: %s" %font_size)
    except NoSuchElementException:
        print("No se encontró font-size")
        return

def get_dimensiones(item, elemento, url):
    height = item.rect["height"]
    width = item.rect["width"]
    ubicacion_y = item.location["y"]
    ubicacion_x = item.location["x"]
    if elemento == "header":
        cursor.execute("UPDATE Test.dbo.dimensiones SET
header_height = ?, header_width = ? WHERE url = ?", height, width,
url)
        print("Se procesó dimensiones para %s" %elemento)
        cursor.execute("UPDATE Test.dbo.ubicacion SET header_x =
?, header_y = ? WHERE url = ?", ubicacion_x, ubicacion_y, url)
        print("Se procesó ubicacion para %s" %elemento)
    if elemento == "footer":
        cursor.execute("UPDATE Test.dbo.dimensiones SET
footer_height = ?, footer_width = ? WHERE url = ?", height, width,
url)
        print("Se procesó dimensiones para %s" %elemento)
        cursor.execute("UPDATE Test.dbo.ubicacion SET footer_x =
?, footer_y = ? WHERE url = ?", ubicacion_x, ubicacion_y, url)
        print("Se procesó ubicacion para %s" %elemento)
    if elemento == "search":
        cursor.execute("UPDATE Test.dbo.dimensiones SET
search_height = ?, search_width = ? WHERE url = ?", height, width,
url)
        print("Se procesó dimensiones para %s" %elemento)
        cursor.execute("UPDATE Test.dbo.ubicacion SET search_x =
?, search_y = ? WHERE url = ?", ubicacion_x, ubicacion_y, url)
        print("Se procesó ubicacion para %s" %elemento)
    if elemento == "logo":
        cursor.execute("UPDATE Test.dbo.dimensiones SET
logo_height = ?, logo_width = ? WHERE url = ?", height, width, url)
        print("Se procesó dimensiones para %s" %elemento)
        cursor.execute("UPDATE Test.dbo.ubicacion SET logo_x = ?,
logo_y = ? WHERE url = ?", ubicacion_x, ubicacion_y, url)
```

```
        print("Se procesó ubicación para %s" %elemento)

main()
conexion.commit()
driver.close()
driver.quit()
```

2.3 Creación de parámetros

```
import pyodbc
import statistics
from statistics import mode

conexion = pyodbc.connect('Driver={SQL Server};'
    'Server=LAPTOP-CH0B4N23\SQLEXPRESS;'
    'Database=Test;'
    'Trusted_Connection=yes;')

cursor = conexion.cursor()

elementos = ["header", "footer", "logo", "searchbox", "background",
"letra", "font_size", "loadtime", "dimensiones", "ubicacion",
"ubicacion_media"]

def puntuar(numero):
    if float(numero) == 0 or len(numero) <= 3:
        return numero
    return f"{int(numero):,}".replace(',', ' .')

def get_ganador(num, uno, dos, tres):

    if num == uno:
        return "tag"
    elif num == dos:
        return "ide"
    elif num == tres:
        return "class"

def mas_comun(lista):
    return (mode(lista))

def make_param(elemento):

    if elemento == "background" or elemento == "letra" or elemento
    == "font_size":
        lista = []
        for row in cursor:
            lista.append(row[0])
        final = mas_comun(lista)
        cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", final, elemento)
        conexion.commit()
        return

    i = 0
    tags = 0
    ides = 0
    classes = 0
    for row in cursor:
        i += 1
        if int(row[1]) == 1:
            tags += 1
```

```
        if int(row[2]) == 1:
            ides += 1
        if int(row[3]) == 1:
            classes += 1

num_ganador = max(tags, ides, classes)
ganador = get_ganador(num_ganador, tags, ides, classes)

print("Elemento revisado: %s" %elemento)
print("Numero de paginas revisadas: " + str(i))
print("Numero de tag names: " + str(tags))
print("Numero de ides: " + str(ides))
print("Numero de class names: " + str(classes))
print("El ganador es: " + ganador + " con " + str(num_ganador) +
" ocurrencias")

        cursor.execute("UPDATE Test.dbo.parametros SET metodo = ? WHERE
item = ? ", ganador, elemento)

def promedio(elemento):
    total = 0
    i = 0
    for row in cursor:
        total += float(row[1])
        i += 1
    promedio = round((total/i),2)
    print("Elemento revisado: %s" %elemento)
    print("Numero de paginas revisadas: " + str(i))
    print("El promedio es: %s segundos" %promedio)
    cursor.execute("UPDATE Test.dbo.parametros SET metodo = ? WHERE
item = ? ", str(promedio).replace('.', ','), elemento)

for elemento in elementos:
    if (elemento == "header"):
        cursor.execute('SELECT * FROM Test.dbo.header')
        make_param(elemento)
    if (elemento == "footer"):
        cursor.execute('SELECT * FROM Test.dbo.footer')
        make_param(elemento)
    if (elemento == "logo"):
        cursor.execute('SELECT * FROM Test.dbo.logo')
        make_param(elemento)
    if (elemento == "searchbox"):
        cursor.execute('SELECT * FROM Test.dbo.searchbox')
        make_param(elemento)
    if (elemento == "background"):
        cursor.execute('SELECT background FROM Test.dbo.color')
        make_param(elemento)
    if (elemento == "letra"):
        cursor.execute('SELECT letra FROM Test.dbo.color')
        make_param(elemento)
    if (elemento == "font_size"):
        cursor.execute('SELECT font_size FROM Test.dbo.font_size')
        make_param(elemento)
```

```
if (elemento == "loadtime"):
    cursor.execute('SELECT * FROM Test.dbo.loadtime')
    promedio(elemento)
if (elemento == "dimensiones"):
    page_height = 0
    page_width = 0
    header_height = 0
    header_width = 0
    footer_height = 0
    footer_width = 0
    search_height = 0
    search_width = 0
    logo_height = 0
    logo_width = 0
    page_height_counter = 0
    page_width_counter = 0
    header_height_counter = 0
    header_width_counter = 0
    footer_height_counter = 0
    footer_width_counter = 0
    search_height_counter = 0
    search_width_counter = 0
    logo_height_counter = 0
    logo_width_counter = 0
    cursor.execute('SELECT * FROM Test.dbo.dimensiones')
    for row in cursor:
        if row[1] is not None:
            page_height += float(row[1])
            page_height_counter += 1
        if row[2] is not None:
            page_width += float(row[2])
            page_width_counter += 1
        if row[3] is not None:
            header_height += float(row[3])
            header_height_counter += 1
        if row[4] is not None:
            header_width += float(row[4])
            header_width_counter += 1
        if row[5] is not None:
            footer_height += float(row[5])
            footer_height_counter += 1
        if row[6] is not None:
            footer_width += float(row[6])
            footer_width_counter += 1
        if row[7] is not None:
            search_height += float(row[7])
            search_height_counter += 1
        if row[8] is not None:
            search_width += float(row[8])
            search_width_counter += 1
        if row[9] is not None:
            logo_height += float(row[9])
            logo_height_counter += 1
        if row[10] is not None:
```

```
        logo_width += float(row[10])
        logo_width_counter += 1
    page_height = round(page_height/page_height_counter)
    page_width = round(page_width/page_width_counter)
    header_height = round(header_height/header_height_counter)
    header_width = round(header_width/header_width_counter)
    footer_height = round(footer_height/footer_height_counter)
    footer_width = round(footer_width/footer_width_counter)
    search_height = round(search_height/search_height_counter)
    search_width = round(search_width/search_width_counter)
    logo_height = round(logo_height/logo_height_counter)
    logo_width = round(logo_width/logo_width_counter)
    cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", f"{page_height:,}".replace(',','.'), "page_height")
    cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", f"{page_width:,}".replace(',','.'), "page_width")
    cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", f"{header_height:,}".replace(',','.'),
"header_height")
        cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", f"{header_width:,}".replace(',','.'),
"header_width")
        cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", f"{footer_height:,}".replace(',','.'),
"footer_height")
        cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", f"{footer_width:,}".replace(',','.'),
"footer_width")
        cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", f"{search_height:,}".replace(',','.'),
"search_height")
        cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", f"{search_width:,}".replace(',','.'),
"search_width")
        cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", f"{logo_height:,}".replace(',','.'), "logo_height")
        cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", f"{logo_width:,}".replace(',','.'), "logo_width")
    if (elemento == "ubicacion"):
        header_x = 0
        header_y = 0
        footer_x = 0
        footer_y = 0
        search_x = 0
        search_y = 0
        logo_x = 0
        logo_y = 0
        header_x_counter = 0
        header_y_counter = 0
        footer_x_counter = 0
        footer_y_counter = 0
        search_x_counter = 0
        search_y_counter = 0
        logo_x_counter = 0
```

```

logo_y_counter = 0
cursor.execute('SELECT * FROM Test.dbo.ubicacion')
for row in cursor:
    if row[1] is not None:
        header_x += float(row[1])
        header_x_counter += 1
    if row[2] is not None:
        header_y += float(row[2])
        header_y_counter += 1
    if row[3] is not None:
        footer_x += float(row[3])
        footer_x_counter += 1
    if row[4] is not None:
        footer_y += float(row[4])
        footer_y_counter += 1
    if row[5] is not None:
        search_x += float(row[5])
        search_x_counter += 1
    if row[6] is not None:
        search_y += float(row[6])
        search_y_counter += 1
    if row[7] is not None:
        logo_x += float(row[7])
        logo_x_counter += 1
    if row[8] is not None:
        logo_y += float(row[8])
        logo_y_counter += 1
header_x = round(header_x/header_x_counter)
header_y = round(header_y/header_y_counter)
footer_x = round(footer_x/footer_x_counter)
footer_y = round(footer_y/footer_y_counter)
search_x = round(search_x/search_x_counter)
search_y = round(search_y/search_y_counter)
logo_x = round(logo_x/logo_x_counter)
logo_y = round(logo_y/logo_y_counter)
cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", f"{header_x:,}".replace(',','.'), "header_x")
cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", f"{header_y:,}".replace(',','.'), "header_y")
cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", f"{footer_x:,}".replace(',','.'), "footer_x")
cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", f"{footer_y:,}".replace(',','.'), "footer_y")
cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", f"{search_x:,}".replace(',','.'), "search_x")
cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", f"{search_y:,}".replace(',','.'), "search_y")
cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", f"{logo_x:,}".replace(',','.'), "logo_x")
cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", f"{logo_y:,}".replace(',','.'), "logo_y")
if elemento == "ubicacion_media":
    cursor.execute('SELECT * FROM Test.dbo.ubicacion')
    header_x_media = []

```

```
header_y_media = []
footer_x_media = []
footer_y_media = []
search_x_media = []
search_y_media = []
logo_x_media = []
logo_y_media = []
for row in cursor:
    if row[1] is not None:
        header_x_media.append(row[1])
    if row[2] is not None:
        header_y_media.append(row[2])
    if row[3] is not None:
        footer_x_media.append(row[3])
    if row[4] is not None:
        footer_y_media.append(row[4])
    if row[5] is not None:
        search_x_media.append(row[5])
    if row[6] is not None:
        search_y_media.append(row[6])
    if row[7] is not None:
        logo_x_media.append(row[7])
    if row[8] is not None:
        logo_y_media.append(row[8])
header_x_media = mas_comun(header_x_media)
header_y_media = mas_comun(header_y_media)
footer_x_media = mas_comun(footer_x_media)
footer_y_media = mas_comun(footer_y_media)
search_x_media = mas_comun(search_x_media)
search_y_media = mas_comun(search_y_media)
logo_x_media = mas_comun(logo_x_media)
logo_y_media = mas_comun(logo_y_media)
print(footer_y_media)
cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", puntuar(header_x_media), "header_x_media")
cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", puntuar(header_y_media), "header_y_media")
cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", puntuar(footer_x_media), "footer_x_media")
cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", puntuar(footer_y_media), "footer_y_media")
cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", puntuar(search_x_media), "search_x_media")
cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", puntuar(search_y_media), "search_y_media")
cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", puntuar(logo_x_media), "logo_x_media")
cursor.execute("UPDATE Test.dbo.parametros SET metodo = ?
WHERE item = ? ", puntuar(logo_y_media), "logo_y_media")
conexion.commit()
```

3 RESULTADO GRÁFICO EVALUACIÓN

Resultados Obtenidos

Página analizada: <https://www.paris.cl>

Tiempo de carga y aspectos generales

Glosa	Medición	Comentario
Tiempo de carga	0,958	Tiempo de carga menor al esperado de 2,67 segundos
Altura de la página	3.604	Página mas alta de lo recomendado. La altura de la página es 3.604 pixeles y La altura recomendado es de no mas de 3 pantallas completas. La altura de la pantalla utilizada es de 1.056 pixeles
Infinite scrolling	Si	La página posee infinite scrolling: 3 scrolls
Ancho de la página	1.903	El ancho de la página es 1.903, el ancho de la pantalla es 1.936 y el ancho recomendado de la página es 1.548
Se encontró h1	Sí	Se encontró correctamente al menos un h1: NAVIDAD PARIS

Header

Glosa	Medición	Comentario
Se encontró header	Si	Se encontró correctamente el header por el tag_name
Altura del header	101	El header es mas pequeño de lo esperado: 192.203 < 244.120
Ancho del header	1.903	
Coordenada 'x' del header	0	El header se encuentra en la posición esperada: (0, 0) == (0,0)
Coordenada 'y' del header	0	

Footer

Glosa	Medición	Comentario
Se encontró footer	Si	Se encontró correctamente el footer por el tag_name
Altura del footer	737	El footer es mas grande de lo esperado: 1.402.511 > 735.600
Ancho del footer	1.903	
Coordenada 'x' del footer	0	El footer no se encuentra en la posición esperada, dado que se espera que su coordenada en 'x' sea 0 y su coordenada en 'y' sea la resta entre la altura de la página de 14.043 pixeles y su altura de 737 pixeles: 13.306
Coordenada 'y' del footer	13.292	

Barra de búsqueda

Glosa	Medición	Comentario
Se encontró search	No	No se encontró el search buscando por class_name

Logo

Glosa	Medición	Comentario
Se encontró logo	Si	Se encontró correctamente el logo por el class_name
Altura del logo	74	Logo mas grande de lo esperado: 14.652 > 14.400
Ancho del logo	198	
Coordenada 'x' del logo	15	El logo no se encuentra en la posición esperada (Superior izquierda)
Coordenada 'y' del logo	27	

Letra y Color

Glosa	Medición	Comentario
Color de fondo	rgba(255, 255, 255, 1)	El color de fondo es igual al esperado: rgba(255, 255, 255, 1)
Color de la letra	rgba(26, 26, 26, 1)	El color de la letra no es igual al esperado: rgba(0, 0, 0, 1)
Tamaño de la letra	10px	El tamaño de la letra es menor al esperado: 16px
Fuente del texto	"Open Sans", sans-serif	Se encontró la fuente de texto esperada dentro de la familia de fuentes
Ajuste de letra correcto	text-size-adjust: 100%	Se encontró 1 propiedad que fija bien el tamaño de la letra: text-size-adjust: 100%
Ajuste de letra incorrecto		No se encontró propiedades que fijen mal el tamaño de la letra

Links

Glosa	Medición	Comentario
Nr de links	225	La página posee 225 links
Links correctos	0	Ningún link cumple con lo esperado
Links incorrectos	225	
Links subrayados	0	El numero de links subrayados es: 0
Links azules	0	Los links que cumplen con el color azul son: 0
Links suficientemente distintos	186	Los links que tienen un color distinto al azul pero suficientemente distinto al color de la letra son: 186

Publicidad y Media

Glosa	Medición	Comentario
Nr de videos	0	No se detectó videos que se reprodujeran automaticamente
Nr de audios	0	No se detectó pistas de audio que se reprodujeran automaticamente
Nr de anuncios	0	No se encontró anuncios en la página

Parámetros

4 RESULTADO GRÁFICO PARÁMETROS

Parámetros Utilizados

Comentarios Generales

Los parámetros mostrados en esta página fueron obtenidos mediante un software que analiza las 50 páginas chilenas mas visitadas. El software extrae de las páginas todo lo necesario y relevante para la creación de una parámetros útil y correcta. El objetivo de crear esta parámetros es tener un punto de comparación para los resultados obtenidos en la Evaluación y poder entregar una respuesta que sirva al usuario, en términos de funcionalidad y diseño web.

A continuación, se explica la manera de mostrar cada Parámetro:

- Las medidas de altura y ancho están en pixeles (px).
- Las coordenadas 'x' e 'y' están en pixeles (px).
- Las medidas de tiempo están en segundos.
- Los métodos de búsqueda para header, footer, barra de búsqueda y logo pueden ser una de tres opciones: tag_name, id y class_name.

Lista de parámetros obtenidos mediante análisis previo

Header

Glosa	Parámetro
Método de búsqueda	tag
Altura del header	136
Ancho del header	1.795
Coordenada 'x' (Promedio)	28
Coordenada 'y' (Promedio)	18
Coordenada 'x' (Media)	0
Coordenada 'y' (Media)	0

Footer

Glosa	Parámetro
Método de búsqueda	tag
Altura del footer	400
Ancho del footer	1.839
Coordenada 'x' (Promedio)	9
Coordenada 'y' (Promedio)	6.695
Coordenada 'x' (Media)	0
Coordenada 'y' (Media)	27.483

Barra de búsqueda

Glosa	Parámetro
Método de búsqueda	class
Altura de la barra de búsqueda	33
Ancho de la barra de búsqueda	219
Coordenada 'x' (Promedio)	480
Coordenada 'y' (Promedio)	84
Coordenada 'x' (Media)	0
Coordenada 'y' (Media)	0

Logo

Glosa	Parámetro
Método de búsqueda	class
Altura del logo	55
Ancho del logo	190
Coordenada 'x' (Promedio)	465
Coordenada 'y' (Promedio)	1.153
Coordenada 'x' (Media)	382
Coordenada 'y' (Media)	1

Esconder Comentarios

Los parámetros que corresponden a métodos de búsqueda fueron obtenidos extrayendo la forma mas frecuente de uso para el ítem correspondiente. Si un ítem debe ser buscado por el tag_name, significa que en la mayoría de las páginas utilizadas en el análisis, la palabra usada para buscar ese ítem se encontró en el tag_name. En primer lugar se intenta encontrar el ítem buscando por tag_name, al no lograrlo con el tag_name se intenta por id y de no encontrarse con ninguna de las opciones anteriores, se recurre al class_name.

En esta instancia es relevante mencionar, que la búsqueda de tag_name, id y class_name se realiza comparando exactamente el nombre del tag html con lo buscado. En otras palabras, si el id buscado es "header" y se encuentra un "head" el sistema no lo reconoce.

Las medidas de altura y ancho fueron obtenidas promediando todas las medidas obtenidas en el análisis. Se considera que los promedios para estas dos medidas entregan suficiente información para ser usados como parámetro, dado que la muestra es lo suficientemente grande. Para las coordenadas se muestra el promedio por la misma razón pero además se muestra la media. Lo anterior dado que luego de evaluar el set de datos se observó que las medidas obtenidas en el análisis eran predominantemente de un mismo valor, con algunos valores de borde que no tienen mucha relevancia y no permiten que el promedio entregue información relevante. Se muestran los dos valores para un mayor entendimiento del set de datos.

Tiempo de carga y dimensiones generales

Glosa	Parámetro
Tiempo de carga	2.67
Altura de la página	7.028
Ancho de la página	1.884

Esconder Comentarios

El tiempo de carga mostrado corresponde al promedio de los tiempos de carga obtenidos en el análisis. Para efectos de lograr una medición exhaustiva, se decidió dejar un límite máximo de tiempo de 60 segundos, antes de que el sistema deje de intentar ingresar a la página. La convención es que el tiempo de carga no puede exceder los 10 segundos, por lo que si el medido es menor a esto se considerará como bueno, mientras que si es igual o menor al parámetro obtenido con el análisis, se considerará muy bueno.

Las dimensiones de la página son obtenidas de la misma forma, promediando las medidas encontradas en todas las páginas usadas para el análisis.

Letra y Color

Glosa	Parámetro
Color de fondo	rgba(255, 255, 255, 1)
Color de la letra	rgba(0, 0, 0, 1)
Tamaño de la letra	16px
Fuente del texto	Sans-serif

Esconder Comentarios

Los 3 parámetros anteriores fueron obtenidos calculando la moda de la muestra. Dado que los resultados de colores y tamaño de letra son muy variados y el promedio no entrega una información útil, se decidió utilizar como parámetro el resultado con mas repeticiones.

[Volver](#)