UNIVERSIDAD DE ATACAMA

FACULTAD DE INGENIERÍA
DEPTO. DE ING. INFORMÁTICA Y CIENCIAS DE LA COMPUTACIÓN

ESTUDIO EXPLORATORIO SOBRE LA VERIFICACIÓN AUTOMÁTICA DE LA USABILIDAD DE SISTEMAS DE SOFTWARE

Trabajo de titulación presentado como parte de los requisitos para optar al título de Ingeniero Civil en Computación e Informática
(Modalidad de Apoyo a la Investigación)

Profesor Guía: John W. Castro Llanos

Ignacio Jesús Garnica Vilches

Copiapó, Mayo 2022, Chile.

UNIVERSIDAD
DE ATACAMA

FACULTAD DE INGENIERÍA
DEPTO. DE ING. INFORMÁTICA Y CIENCIAS DE LA COMPUTACIÓN

ESTUDIO EXPLORATORIO SOBRE LA VERIFICACIÓN AUTOMÁTICA DE LA USABILIDAD DE SISTEMAS DE SOFTWARE

Trabajo de titulación presentado como parte de los requisitos para optar al título de Ingeniero Civil en Computación e Informática
(Modalidad de Apoyo a la Investigación)

Profesor Guía:
John W. Castro Llanos

Miembros del Comité:
Dante Carrizo Moreno
Nahur Meléndez Araya

Ignacio Jesús Garnica Vilches

Copiapó, Mayo 2022, Chile.

Este trabajo de titulación presentado se la dedico a mi familia, compañeros y amigos de universidad. También a todos los que me ayudaron durante el proceso de educación superior.

Agradecimientos

Agradezco a mi familia por ayudarme y apoyarme durante todo este proceso universitario. Agradezco al Departamento de Ingeniería en Informática y Ciencias de la Computación por brindar los espacios necesarios para llevar a cabo mi proceso estudiantil. Agradezco a sus profesores y en especial a mi profesor guía, John Castro, por ayudarme y brindarme su conocimiento durante todo el proceso de investigación y confección de este trabajo. Además, agradezco al Ministerio de Educación por brindar apoyo económico gracias a los Fondos Basales para Alumnos Ayudantes, correspondiente al proyecto ATA1899. Por último, agradezco especialmente a mis compañeros y amigos de universidad, ya que sin ellos y su apoyo no estaría donde estoy ahora.

Resumen

La usabilidad es una de las medidas mas importantes a la hora de garantizar la calidad del software. Para que ésta pueda ser medida, se debe realizar una evaluación de la usabilidad, pero esto trae ciertas desventajas ya que requiere de un costo elevado de tiempo y dinero, necesita de expertos de usabilidad y a pesar de tener experiencia, estos expertos siempre entregarán un componente de subjetividad en sus evaluaciones. Es por esto que se busca utilizar herramientas que permitan la evaluación automática de la usabilidad. Con el uso de éstas, se busca mitigar estas desventajas. Lamentablemente, no existe un cuerpo de conocimiento que las agrupe. Por tal razón, en el presente trabajo se realiza un Mapeo Sistemático de Literatura (SMS o *Systematic Mapping Study* por sus siglas en inglés) con el objetivo de identificar el panorama actual de las herramientas que permitan la evaluación automática de la usabilidad abarcando la literatura existente, generando así un cuerpo de conocimiento de libre acceso para que cualquiera que quiera realizar una evaluación de la usabilidad tenga a su disposición una serie de herramientas. Este estudio abordó la literatura entre el 2016 y el 2021 considerando las bases de datos Scopus, IEEE Xplore y Web of Science, encontrándose 14 herramientas que permiten apoyar la evaluación de la usabilidad, siendo estas explicadas y clasificándose en categorías para aclarar su alcance en materia de evaluación de la usabilidad. Si bien las herramientas relacionadas al objetivo de esta investigación no reemplazan directamente una evaluación de usabilidad manual, la apoyan de manera significativa, reduciendo costos y permitiendo la detección de errores de usabilidad que a veces son pasados por alto en una evaluación manual de la usabilidad.

Palabras clave: usabilidad, evaluación, herramienta, automatizado, automática.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

El trabajo de investigación que se presenta en este documento se enmarca en las áreas de la usabilidad, las técnicas de la evaluación de la usabilidad y las herramientas que permiten evaluar la usabilidad de forma automatizada, apoyando las técnicas de la evaluación de la usabilidad. En primer lugar, se explica el contexto del que surge la investigación, considerando los puntos anteriormente mencionados. En segundo lugar, se presentan las preguntas de investigación con el objetivo de desarrollar este trabajo y darles una respuesta. En tercer lugar, se detallan los objetivos, general y específicos, de esta investigación. Por último, se describe la estructura de este trabajo, explicando a grandes rasgos de que se trata cada capítulo.

1.1. Visión General de la Investigación

Actualmente, existe un crecimiento de los sistemas software desarrollados. Esto ha ocasionado que cada vez se exija mayor calidad de los sistemas, la cual se puede asegurar con ciertas medidas y métodos estandarizados por medio de diferentes actividades y técnicas. Una de las medidas más importantes a la hora de desarrollar un sistema software es la usabilidad [Nielsen, 1994].

La usabilidad se considera como la medida en que usuarios utilizan un sistema, producto o servicio de eficacia y satisfacción, dado un contexto de uso [ISO, 2018]. Ésta también puede estar ligada con la aceptabilidad, por parte de los usuarios, de un sistema específico, considerando que este sea lo suficientemente bueno como para satisfacer las necesidades de los usuarios [Nielsen, 1994]. Para garantizar que estas

exigencias se cumplan, los sistemas desarrollados deben someterse a una evaluación de la usabilidad.

A lo largo de la historia se han desarrollado técnicas, métodos, pautas y métricas para realizar evaluaciones de usabilidad que permitan medir el nivel de usabilidad de un sistema desarrollado [Ferré, 2005; Ivory y Hearst, 2001; Nielsen, 1994]. El reto de desarrollar software usable es apoyado por la evaluación de la usabilidad, la cuál permite medir la usabilidad de un sistema de software mediante el uso de métodos y técnicas que faciliten dicha tarea [Ferré, 2005; Ivory y Hearst, 2001]. Para determinar qué tan usable es un sistema se deben consideran factores de la interfaz de usuario (IU) como los colores usados, la disposición de elementos, la interacción con el usuario, cantidad y calidad de texto presentado, entre otros [Nielsen, 1994].

A pesar de la importancia de la evaluación de la usabilidad para cualquier sistema software desarrollado, ésta presenta ciertas desventajas, como un alto coste de tiempo y presupuesto dadas sus características. Además, algunas técnicas relacionadas con la evaluación de la usabilidad, para ser implementadas, necesitan como mínimo un experto en materia de usabilidad [Ferré, 2005; Ivory y Hearst, 2001]. Este experto puede guiarse con pautas, métricas y heurísticas para apoyar la labor de la evaluación de la usabilidad, pero a pesar de todo, este experto evaluador siempre entregará un cierto nivel de subjetividad en su análisis [Ivory y Hearst, 2001; Marenkov *et al.*, 2018]. Si bien estas desventajas pueden ser desalentadoras, a pesar de los beneficios que entregan considerando el producto de software finalizado, pueden ser mitigadas implementando herramientas que apoyan la evaluación de la usabilidad [Fabo y Durikovic, 2012; Federici *et al.*, 2018; Grigera *et al.*, 2017b; Marenkov *et al.*, 2018].

Las herramientas para la evaluación de la usabilidad son sistemas que apoyan esta tarea. Existen muchas de estas que benefician directamente las actividades de la evaluación de la usabilidad de forma automatizada, permitiendo, por ejemplo, durante una prueba de usabilidad almacenar datos de registro de usuarios como: (i) la pulsación de las teclas, (ii) los clics realizados con el ratón y (iii) las distancias recorridas por el puntero de éste, entre otras más. Estas herramientas permiten en algunos casos analizar los datos recolectados con el fin de entregar retroalimentación a los desarrolladores y a los expertos de usabilidad, entregando información de errores de usabilidad y, dependiendo de la herramienta, corrigiéndolos automática-

mente [Fabo y Durikovic, 2012; Federici *et al.*, 2018; Grigera *et al.*, 2017b; Liyanage y Vidanage, 2016; Marenkov *et al.*, 2018].

Actualmente, existe una variedad de estas herramientas gracias a investigaciones preliminares. A pesar de esto, no se detectó la existencia de algún estudio o trabajo científico que agrupe y clasifique las herramientas. Lo que se busca es, mediante una investigación de la literatura existente, conocer el panorama general de las herramientas para la evaluación automática de la usabilidad, identificar dichas herramientas y agruparlas en un solo cuerpo de conocimiento con el fin de atacar directamente las desventajas más importantes que presenta la implementación de una evaluación de la usabilidad.

1.2. Preguntas de Investigación

Para guiar esta investigación y de acuerdo al contexto de ésta, se buscará responder a las siguientes preguntas de investigación (PI):

PI1 ¿Cuáles son las herramientas automatizadas que apoyan la evaluación de la usabilidad?

PI2 ¿Cuáles son las técnicas relacionadas con la evaluación de la usabilidad que se benefician de las herramientas automatizadas?

PI3 ¿Cuáles son los problemas y retos existentes del uso de herramientas automatizadas para la evaluación de la usabilidad?

PI4 ¿Cómo se pueden clasificar las herramientas automatizadas para la evaluación de la usabilidad?

1.3. Objetivos

La usabilidad es un factor importante a la hora de desarrollar software y esta investigación tiene en cuenta las herramientas que permiten apoyar a la evaluación de la usabilidad. A continuación, detallamos el objetivo general y los objetivos específicos planteados en esta investigación.

1.3.1. Objetivo General

Conocer el panorama general de las herramientas para la evaluación automática de la usabilidad.

1.3.2. Objetivos Específicos

Considerando el objetivo general, para cumplir con este, consideraremos los siguientes objetivos específicos:

a) Identificar las herramientas automatizadas para la evaluación de la usabilidad.

b) Determinar las técnicas que se benefician de las herramientas automatizadas que apoyan la evaluación de la usabilidad.

c) Identificar los problemas y retos existentes del uso de herramientas automatizadas.

d) Clasificar las herramientas automatizadas según las técnicas de evaluación de la usabilidad.

1.4. Estructura del Trabajo

Este trabajo busca realizar un mapeo sistemático de literatura sobre las herramientas existentes que apoyan la evaluación de la usabilidad de sistemas software. Se divide en los siguientes capítulos:

- El presente es el **primer capítulo** e introduce el trabajo de investigación, además de presentar las PI y sus objetivos.
- El **segundo capítulo** presenta un marco teórico de las temáticas relacionadas con el ámbito de investigación. Se explica la usabilidad como concepto, la evaluación de la usabilidad, las técnicas de evaluación de la usabilidad y en qué consisten las herramientas que apoyan la evaluación de la usabilidad.
- El **tercer capítulo** presenta trabajos relacionados, mostrando sus enfoques y cómo se relacionan de alguna forma con la investigación presentada en este trabajo.

- El **cuarto capítulo** presenta la metodología de investigación usada para realizar este trabajo. En este caso, se realizó un mapeo sistemático de literatura.
- En el **quinto capítulo** se realiza la síntesis de acuerdo con los resultados obtenidos de la investigación realizada. Se busca además responder a las PI y cumplir con los objetivos generales y específicos.
- En el **sexto capítulo** se discute acerca de los resultados obtenidos junto con las amenazas a la validez correspondientes.
- En el **séptimo capítulo** se presentan las conclusiones y trabajos futuros, siguiendo la línea de la investigación presentada en este trabajo.
- En el **Apéndice A** se presentan las palabras obtenidas del Grupo de Control para la creación de la cadena de búsqueda que será usada para obtener los estudios primarios.
- En el **Apéndice B** se listan los estudios primarios obtenidos a partir del mapeo sistemático de literatura.
- En el **Apéndice C** se reportan las técnicas relacionadas con la evaluación de la usabilidad.
- En el **Apéndice D** se adjunta la publicación derivada en la HCII 2022 (ver sección 1.5).
- Por último, en el **Apéndice E** se explica con detalle las herramientas que apoyan la evaluación de la usabilidad de forma automática, explicando su funcionamiento y enfoques.

1.5. Publicación Derivada

A raíz del presente Trabajo de Titulación, se ha realizado la siguiente publicación en una conferencia especializada en el área de la Interacción Persona-Ordenador:

- Automated Tools for Usability Evaluation: A Systematic Mapping Study. *24th International Conference on Human-Computer Interaction* (HCII'22), july 2022. Virtual Conference, 1-19. Indexado en: Scopus.

Capítulo 2

Marco Teórico

El presente capítulo tiene por objetivo describir aspectos teóricos relacionados con el problema de investigación descrito anteriormente. En primer lugar, se explicará la usabilidad como concepto general en el contexto de software. En segundo lugar, se explicará la evaluación de la usabilidad que, de acuerdo a la definición de usabilidad, busca medirla con el fin de determinar qué tan usable es un sistema software particular. En tercer lugar, se explicarán algunas de las técnicas relacionadas con la evaluación de la usabilidad, las cuales apoyan al proceso de evaluación correspondiente. Por último, y en el contexto de esta investigación, se detallará en qué consisten las herramientas para la evaluación automática de la usabilidad, explicando el porqué de su implementación y las ventajas que conlleva integrarlas en la evaluación de la usabilidad.

2.1. Usabilidad

La usabilidad es una característica de la calidad del software utilizada en la mayoría de clasificaciones [Losana *et al.*, 2021]. Se define como la medida en que usuarios específicos pueden utilizar un sistema, producto o servicio para lograr objetivos específicos con eficacia y satisfacción en un contexto de uso específico [ISO, 2018]. Esta es una definición más amplia del concepto de usabilidad, aunque no del todo precisa para ámbitos relacionados con la ingeniería de software. Nielsen expone que la usabilidad está ligada con la aceptabilidad, por parte de los usuarios, de un sistema específico, esperando que éste sea lo suficientemente bueno como para satisfacer las necesidades y requisitos de estos usuarios [Nielsen, 1994]. Nielsen también

menciona que no se debe considerar la usabilidad como un aspecto unidimensional, porque tiene varios componentes relacionados con los siguientes atributos:

1. Capacidad de aprendizaje: el sistema debe ser fácil de aprender para el usuario.
2. Eficiencia: el sistema debe tener un uso eficiente.
3. Memorable: el sistema debe ser fácilmente recordable por el usuario.
4. Errores: el sistema debe tener un bajo ratio de errores y que estos sean fácilmente corregibles para el usuario.
5. Satisfacción: el sistema debe ser satisfactorio en su uso por el usuario.

Un sistema software se considera "usable" cuando este cumple satisfactoriamente con estos atributos. Para garantizar esto, el sistema debe someterse a una evaluación de la usabilidad.

2.2. Evaluación de Usabilidad

La usabilidad del software ya no es un lujo, sino un determinante básico de la aceptación de los sistemas desarrollados [Ferré *et al.*, 2002a,b]. Para garantizar los estándares de calidad de usabilidad, éstos se miden mediante la evaluación de la usabilidad, que busca determinar qué tan usable es un sistema software considerando para esto una serie de técnicas, pautas, heurísticas y metodologías. La evaluación de la usabilidad permite detectar problemas en la interacción del usuario con el sistema software, con el fin de corregirlos y mejorar así la experiencia del usuario final con el uso del software desarrollado. En general, esta evaluación es realizada por un experto en el área de usabilidad, aunque dependiendo del enfoque y la rigurosidad puede ser realizada por un desarrollador de software [Grigera *et al.*, 2017a]. Con el tiempo, se han desarrollado herramientas que permiten asistir a la evaluación de la usabilidad en sus actividades principales.

Es importante considerar que los aspectos de usabilidad y la evaluación de los mismos están ligados a su plataforma. La evaluación la de usabilidad para un sistema software de escritorio puede ser distinta y con distintos enfoques que una orientada a aplicaciones para móviles, así como también ésta se diferenciaría con la evaluación realizada para una aplicación orientada a web. Considerar el entorno en el cual se

encuentra el sistema software que se quiere evaluar determina las técnicas, pautas, heurísticas y metodologías que se tendrán en cuenta a la hora de realizar la evaluación de la usabilidad [Nielsen, 1994].

En general, los métodos utilizados para la evaluación de la usabilidad comprenden tres actividades: captura de datos de usabilidad, análisis de estos datos y crítica, considerando además propuestas de mejora para los problemas de usabilidad identificados. Los métodos que se usan para realizar la evaluación de la usabilidad de un sistema software se basan en los siguientes enfoques [Ivory y Hearst, 2001]:

a) **Métodos de pruebas de usabilidad**: usuarios reales prueban la IU brindando datos, para luego ser analizados por los expertos.

b) **Métodos de inspección**: un evaluador inspecciona los aspectos de usabilidad del diseño de la IU comparándolo con una colección de pautas. Este tipo de evaluación depende del juicio del evaluador.

c) **Métodos de consulta**: similar al primero mencionado, pero éste incluye la retroalimentación de los usuarios a través de encuestas, formularios, cuestionarios, etc.

d) **Métodos de modelado analítico**: complementan métodos como las pruebas de usabilidad y permite a los evaluadores predecir aspectos de usabilidad de forma barata.

e) **Métodos de simulación**: complementa métodos como el anterior. Los modelos generados simulan la interacción de un usuario con el sistema software y reporta resultados de dicha interacción.

Esta clasificación abarca la mayoría de técnicas de evaluación de la usabilidad aplicables a distintos escenarios y contextos, derivados del entorno en el cual se encuentren, como se explicó anteriormente. Además de los métodos, como se explicó en el punto anterior, se deben considerar las distintas técnicas existentes que apoyan al proceso de evaluación de la usabilidad.

2.3. Técnicas de Evaluación de Usabilidad

Existe un amplio abanico de opciones, siempre considerando el contexto y entorno de uso del sistema software. Diversas fuentes proponen técnicas en el ámbito de la

evaluación de la usabilidad. Ferré realiza una recopilación de las técnicas obtenida a partir del estudio de estas fuentes [Ferré, 2005]. Según Ferré, existen tres tipos de técnicas relacionadas con la evaluación de la usabilidad: Evaluación por Expertos, Test de Usabilidad y Estudios de Seguimiento de Sistemas Instalados [Ferré *et al.*, 2002a,b]. A continuación, se explica cada una de ellas:

a) **Evaluación por Expertos**

La Evaluación por Expertos engloba técnicas tales como la evaluación heurística, inspecciones, recorridos cognitivos y recorrido pluralístico. En general, este tipo de técnicas tiene como énfasis el uso de heurísticas y revisión de pautas y guías para determinar y evaluar la usabilidad de un sistema software. Estas actividades son ejecutadas por expertos en materia de usabilidad, que con su experiencia y experticia pueden evaluar la usabilidad de un sistema.

b) **Test de Usabilidad**

El Test de Usabilidad engloba técnicas tales como medición de rendimiento, pensar en voz alta, información post-test, test de usabilidad en laboratorio, test de campo, grabación de video y de audio, registro de uso, evaluación por control remoto y test remoto por video-conferencia. En general, el grupo de técnicas relacionadas con Test de Usabilidad tienen como objetivo que un usuario pruebe el sistema con una serie de metas que debe cumplir, con el fin de entregar resultados que determinen el desempeño de este con el test. Si el usuario tiene un bajo desempeño en las tareas que fueron impuestas para el test de usabilidad, se definirá mala usabilidad para dicho sistema. En caso contrario, se evalúa el sistema como usable. Las técnicas mencionadas sirven para brindar apoyo a esta idea de prueba de usuario. Grabar las acciones del usuario en video y voz, hacer "pensar en voz alta" mientras usa el software para detectar problemas de usabilidad y la información obtenida después del test, son técnicas que apoyan a la evaluación de la usabilidad en este ámbito.

c) **Estudios de Seguimiento de Sistemas Instalados**

En Estudios de Seguimiento de Sistemas Instalados se engloban técnicas tales como la observación directa, cuestionarios y encuestas, entrevistas, *focus group*, registro de uso y retroalimentación del usuario. Entre estas técnicas, también

se denota la participación de usuarios, pero de forma supervisada. Expertos observan el comportamiento de los usuarios ante un sistema software. También se apoyan en datos entregados por entrevistas y retroalimentación del usuario, en general. Además, se puede hacer uso de herramientas que permitan el registro de la actividad del usuario, como las pulsaciones realizadas durante un tiempo determinado, el registro continuo del rendimiento del usuario con respecto a las tareas designadas para la prueba del sistema, los eventos generales de uso como teclas pulsadas o clics en pantalla, etc.

En el Apéndice C se presentará de forma resumida las técnicas de la evaluación de la usabilidad anteriormente mencionadas. Se aprecia en este Apéndice las técnicas que se desprenden de los tres tipos existentes, todo esto según la recopilación hecha por Ferré *et al.* [2002a,b].

2.4. Herramientas para la Evaluación de la Usabilidad

Como se mencionó anteriormente, con el tiempo se han ido desarrollando herramientas que permiten apoyar la evaluación de la usabilidad. Su enfoque es variado, ya que "apoyo" es una palabra amplia. Un sistema software que permita escribir texto puede servir para apoyar la evaluación de la usabilidad, pero no se considerará algo tan básico. El enfoque se centra, dado el contexto de la evaluación de la usabilidad, en herramientas que permitan apoyar las actividades de la evaluación de la usabilidad. Estas actividades son la captura de datos de usabilidad, el análisis de éstos y la crítica [Ivory y Hearst, 2001].

El uso de herramientas en cualquier actividad facilita la tarea para la cuál están destinadas, y en el proceso de evaluación de la usabilidad no es la excepción. Herramientas con este enfoque entregan ventajas significativas, como la reducción del tiempo en la ejecución de la evaluación de la usabilidad. La concepción general de la evaluación de la usabilidad es la de un proceso lento y engorroso, pero esta puede ser corregida mediante un correcto uso de herramientas, lo que reduce el tiempo de la evaluación, generando así reducción de costos en procesos de desarrollo de software, corrección de errores, etc [Grigera *et al.*, 2017b]. Otro aspecto a considerar es que, en general, la evaluación de la usabilidad es realizada por un experto en la materia.

Este nivel de experticia es difícil de suplir, sobre todo cuando se tienen equipos de desarrolladores que no están especializados en materias de usabilidad. La implementación de herramientas que apoyen la evaluación de la usabilidad brindando colecciones de pautas, heurísticas, y técnicas ayudan en gran medida en estos procesos, en especial para gente inexperta, permitiendo que la evaluación la de usabilidad esté disponible para todos [Grigera *et al.*, 2017b]. Existen más ventajas específicas, pero el objetivo general que buscan suplir las herramientas es hacer que el proceso de evaluación de la usabilidad sea más expedito y fácil de realizar.

Un ejemplo de herramienta automática para la evaluación de la usabilidad es Guideliner [Marenkov *et al.*, 2018]. Esta herramienta permite medir la usabilidad de una página web mediante la comparación de pautas con los atributos de sus elementos durante su fase de implementación. De acuerdo a estas pautas, Guideliner detecta errores de usabilidad del sistema analizado, lo que permite a los desarrolladores poder corregirlos durante la fase de desarrollo. Cabe destacar que esta herramienta solo entrega esta información, pero no realiza la corrección de estos errores. Por ejemplo, de acuerdo con estándares de elementos HTML de una página web, Guideliner realiza la comparación con los elementos que se busca analizar y detecta en qué medida estos presentan problemas de usabilidad, indicándolo como *warning*.

Otro ejemplo de herramienta automática para la evaluación de la usabilidad es Kobold [Grigera *et al.*, 2017b]. Esta permite detectar problemas de usabilidad en aplicaciones web, analizando los elementos HTML y CSS que la conforman. Con base en pautas y métricas, esta herramienta es capaz de detectar problemas de usabilidad, corrigiéndolos si es posible o como mínimo recomendando mejoras y entregando sugerencias para la corrección en materias de usabilidad.

En general, las herramientas que permiten la evaluación de la usabilidad de forma automática se comportan de forma similar a las dos explicadas anteriormente. Dependiendo de la herramienta, tendrá funciones de medición de usabilidad, de corrección automática de errores o de apoyo a la labor de la evaluación de la usabilidad, automatizando algunas de sus actividades.

Capítulo 3

Trabajos Relacionados

En este capítulo se reportan los trabajos relacionados con la temática del presente trabajo de investigación. Hay que destacar que uno de los aspectos que motivó esta investigación fue la falta de trabajos que reportaran el panorama de las herramientas que apoyan la evaluación de la usabilidad de forma automática.

Ivory y Hearst [2001] reportan en su estudio el estado del arte de los métodos de evaluación de la usabilidad, organizados de acuerdo a una taxonomía que enfatiza el papel de la automatización. Ivory y Hearst [2001] centran sus esfuerzos en identificar los aspectos de la automatización de la evaluación de la usabilidad que sean útiles en investigaciones futuras y sugieren nuevas formas de expandir los enfoques existentes para respaldar mejor la evaluación de la usabilidad. Este estudio se interpreta como precursor de los enfoques automatizados que, con el tiempo, se convirtieron en procesos de desarrollo de herramientas que permitan la evaluación automática de la usabilidad. A lo largo de su estudio, se nombran varias herramientas, aunque no tan sofisticadas como las que existen actualmente (considerar el año de publicación de este estudio).

Charfi *et al.* [2014] reportan en su artículo *widgets* basados en evaluación como una contribución para ayudar a los evaluadores en la evaluación temprana de las interfaces de usuario. Se explica que estos *widgets* son capaces de detectar ciertas inconsistencias ergonómicas en el diseño de las interfaces de usuario. Este estudio no realiza un SMS, se centra en exponer los *widgets* que se conocían. Los autores explican los *widgets* en cuanto a funcionalidad y aplicación, además de mostrar una fase experimental donde se prueban éstos. Este estudio muestra estos *widgets* en

un periodo de tiempo anterior al que nosotros consideramos (es decir, entre 2016 y 2021), por lo que no son considerados en nuestro trabajo de investigación.

Bakaev *et al.* [2016] presentan en su artículo una descripción general de los métodos y herramientas dentro de los enfoques tradicionales, semiautomáticos y automáticos para la evaluación de la usabilidad de sitios web. La principal diferencia con nuestro trabajo de investigación, además de que los autores no realizan un SMS, es que Bakaev *et al.* [2016] se enfocan solamente en las herramientas que permiten apoyar la evaluación de la usabilidad automatizada de las interfaces de usuario web, mientras que nosotros centramos nuestros esfuerzos en conocer el panorama general de estas herramientas, ya sean enfocadas a web como para aplicaciones de escritorio y de dispositivos móviles. Las herramientas presentadas en este estudio son descritas pobremente. Nuestro enfoque se centra en reportar las herramientas con un mayor nivel de detalle.

Khasnis *et al.* [2019] exponen en su trabajo de investigación una serie de herramientas que permiten apoyar la evaluación de la usabilidad, explicando en pocas palabras su funcionamiento, ventajas y desventajas de las mismas. Una de las diferencias con nuestra investigación es el detalle con el que explicamos las herramientas que apoyan la evaluación de la usabilidad, lo que se verá en el Apéndice E. Mencionar que el autor no realiza un SMS, como en nuestro caso. Además, Khasnis *et al.* [2019] centran su enfoque en relacionar las herramientas de evaluación automática de usabilidad con métodos de evaluación de la usabilidad. Nuestro enfoque se centra más en relacionar las herramientas reportadas con las técnicas de evaluación de la usabilidad, las cuáles son reportadas por Ferré *et al.* [2002a,b].

Capítulo 4

Metodología de Investigación

El estudio presentado en este trabajo se realizó siguiendo los lineamientos establecidos por Kitchenham *et al.* [2011] para llevar a cabo un Mapeo Sistemático de Literatura o SMS (*Systematic Mapping Study* por sus siglas en inglés). Siguiendo esto, las actividades a realizar de acuerdo con esta metodología son las siguientes:

1. Formular las PI.
2. Definir la estrategia de investigación.
3. Seleccionar los estudios primarios.
4. Extraer los datos de los estudios primarios.
5. Sintetizar los datos extraídos.

La información extraída de los estudios primarios seleccionados debe ser consistente con las PI, y la respuesta a éstas deben destacar las similitudes y diferencias con los resultados de la investigación para facilitar el análisis.

4.1. Preguntas de Investigación

Como se mencionó en el Capítulo 1, dado el contexto entregado y la problemática encontrada, las preguntas que se buscan responder mediante esta investigación son las siguientes:

PI1: ¿Cuáles son las herramientas automatizadas que apoyan la evaluación de la usabilidad?

PI2: ¿Cuáles son las técnicas relacionadas con la evaluación de la usabilidad que se benefician de las herramientas automatizadas?

PI3: ¿Cuáles son los problemas y retos existentes del uso de herramientas automatizadas para la evaluación de la usabilidad?

PI4: ¿Cómo se pueden clasificar las herramientas automatizadas para la evaluación de la usabilidad?

4.2. Definición de la Cadena de Búsqueda

El SMS comienza con la identificación de las palabras clave. Para identificarlas, es necesario encontrar un conjunto de artículos que respondan a las PI. Este conjunto se conoce como Grupo de Control (CG). El CG es un grupo que representa, con la mayor precisión posible, el conjunto conocido de estudios primarios identificados que cumplan con las PI propuestas por el SMS [Zhang *et al.*, 2011]. El CG también sirve como fuente de muestras para perfeccionar las cadenas de búsqueda, además de determinar la sensibilidad de la estrategia de búsqueda definida para el SMS. Se debe tener en consideración que una estrategia de búsqueda altamente sensible recuperará una gran cantidad de resultados, pero muchos de estos pueden ser artículos no deseados y una estrategia de búsqueda más precisa recuperará un número reducido de artículos, pero puede pasar por alto una gran cantidad de estudios que pueden ser de utilidad para la investigación. Es por esto que la conformación de un CG debe tener un equilibrio entre estos dos factores [Zhang *et al.*, 2011].

Para conformar el CG, se realizó una búsqueda tradicional de estudios que tuvieran que ver con el contexto de la investigación y, de acuerdo con la explicación anterior, que respondan a las PI. Como resultado de este proceso de búsqueda, se identificaron seis estudios [Assila *et al.*, 2016; Barra *et al.*, 2019; Federici *et al.*, 2018; Grigera *et al.*, 2017b; Marenkov *et al.*, 2018; Paternò *et al.*, 2017]. Estos se relacionan directamente con la temática de esta investigación y responden a las PI postuladas, es decir, se presentan herramientas que permiten apoyar de forma automática la evaluación de la usabilidad, explicando el funcionamiento de las herramientas tratadas, detalles de experimentación e implementación de la misma.

Previo a construir la cadena de búsqueda, se verifica si los estudios del CG se encuentran en la base de datos Scopus, ya que es la que más estudios alberga.

Dentro de Scopus, se encuentran cinco de los seis que pertenecen al CG, es decir, se encuentran [Barra *et al.*, 2019; Federici *et al.*, 2018; Grigera *et al.*, 2017b; Marenkov *et al.*, 2018; Paternò *et al.*, 2017], ya que [Assila *et al.*, 2016] no se encuentra en Scopus. Con esto, podemos asegurar que trabajar con Scopus es la mejor opción para efectos de la investigación.

Para obtener las palabras clave que servirán para la creación de la cadena de búsqueda, se calcula la frecuencia de palabras presentes en el CG con la ayuda de la herramienta Atlas.ti 9 [Atlas.ti9, 2021]. La Tabla 4.1 presenta un fragmento del listado de palabras obtenido gracias a esta herramienta. El listado completo de palabras puede ser consultado en el Apéndice A. Esta Tabla muestra las veces que una palabra aparece los estudios del CG, por lo que resulta sencillo detectar cuáles son las palabras que más frecuentemente aparecen y en qué estudios. Con esto se obtiene un porcentaje de aparición y su peso asociado. En la Tabla 4.1 se muestra el porcentaje de aparición de palabras y su frecuencia de aparición, además del peso asociado a cada palabra.

Tabla 4.1: Fragmento Tabla de Frecuencias

Palabras	Aparición (%)	Frecuencia	Peso
Usability	100	1156	1
Evaluation	100	577	0.7496
User	100	388	0.6678
Tool	100	240	0.6038
Users	100	224	0.5969
Use	100	150	0.5649
Interface	100	147	0.5636

El peso es calculado basándose en el porcentaje de aparición y la frecuencia y se realiza de la siguiente forma (ver Ecuación 4.1):

$$\text{Peso} = \left(\frac{\%\text{ de Aparición de Palabra}}{\text{Max. \% de Aparición}} + \frac{\text{Frecuencia de Palabra}}{\text{Max. Frecuencia}}\right) \div 2 \qquad (4.1)$$

La importancia de una palabra se representa con su peso. Este valor se encuentra entre el 0 y el 1. Mientras más cercano sea del 1, mayor importancia tendrá y será considerado para la creación de la cadena de búsqueda. Como se muestra en la Tabla 4.1, las palabras que mayor importancia según su peso son *usability*, *evaluation*, *user*, *tool*, *users*, *use*, e *interface*. La lista de frecuencia de palabras completa se

puede encontrar en el Apéndice A. El peso mínimo definido para que una palabra sea considerada es de 0,32.

4.3. Creación de la Cadena de Búsqueda

Una vez se identifican y seleccionan las palabras clave, se construyen varias cadenas de búsqueda con el fin de probarlas y seleccionar la que mejores resultados de búsqueda entreguen. Para la construcción de las cadenas, se consideran cuatro componentes que corresponden a una clasificación de las palabras consideradas. Para definir los componentes, se tomó en cuenta el contexto de esta investigación, que consta de conocer el panorama de las herramientas automáticas que permitan apoyar la evaluación de la usabilidad. Con esto en mente, los componentes resultantes son los siguientes:

a) Componente de "Herramienta" o "*Tool*".

b) Componente de "Automatización".

c) Componente de "Evaluación".

d) Componente de "Usabilidad".

Cada uno de estos componentes se separa con el operador lógico AND. Para separar los sinónimos de las palabras pertenecientes a los componentes mencionados anteriormente, se usa el operador lógico OR. Se crea un total de cuatro cadenas. Estas se muestran en la Tabla 4.2. Se usan estas cadenas para buscar los estudios del CG dentro de la base de datos Scopus. Se debe recordar que cinco de los seis estudios del CG se encuentran en la base de datos Scopus, omitiéndose [Assila *et al.*, 2016], por lo que la cadena que encuentre la mayor cantidad de estos estudios es la que nos servirá para la siguiente etapa del SMS.

En la Tabla 4.2, se muestra la cantidad de estudios encontrados por cada cadena de búsqueda en Scopus. También se muestra cuantos estudios del CG son encontrados por cada cadena de búsqueda. Se puede apreciar que todas las cadenas de búsqueda creadas encuentran los cinco estudios del CG.

Tabla 4.2: Cadenas de Búsqueda Creadas

ID	Cadena de Busqueda	Artículos Encontrados	Artículos Encontrados en CG	Ratio X	Ratio Y	Promedio
1	(usability OR "user experience") AND (evaluation OR testing OR measure OR evaluating OR study OR evaluate OR tests OR assess) AND (tool OR systems OR applications OR tools OR software OR system OR application OR product) AND (automated OR automatic OR automatically OR automating)	2620	5	0.83333	0.00191	0.41762
2	(usability) AND (evaluation OR testing OR measure) AND (tool OR systems OR applications) AND (automated OR automatic OR automatically)	1004	5	0.83333	0.00498	0.41916
3	(usability OR "user experience") AND (evaluation OR testing) AND (tool OR tools OR software OR systems) AND (automated OR automatic)	912	5	0.83333	0.00548	0.41941
4	(usability) AND (evaluation OR testing OR evaluate OR study) AND (tool OR software OR systems) AND (automated OR automatic)	1304	5	0.83333	0.00383	0.41858

Para seleccionar la cadena de búsqueda ganadora fue necesario utilizar indicadores adicionales. Estos indicadores son el ratio X (ver Ecuación 4.2) y el ratio Y (ver Ecuación 4.3), además del Promedio entre ambos (ver Ecuación 4.4), los cuales se muestran a continuación:

$$\text{Ratio X} = \frac{\text{Nro. Artículos Encontrados del CG}}{\text{Total de Artículos del Grupo de Control}} \quad (4.2)$$

$$\text{Ratio Y} = \frac{\text{Nro. Artículos Encontrados del CG}}{\text{Total de Artículos Encontrados por Cadena de Búsqueda}} \quad (4.3)$$

$$\text{Promedio} = \frac{\text{Ratio X} + \text{Ratio Y}}{2} \quad (4.4)$$

En la Tabla 4.2 se puede apreciar que el ratio X se mantiene igual para todas las cadenas de búsqueda. Esto es debido a que con todas las cadenas probadas en la base de datos Scopus se encontraba la misma cantidad de artículos pertenecientes al CG, es decir, cinco de los seis artículos pertenecientes al CG. El ratio Y en cambio muestra ciertas diferencias, ya que este se basa en calcular la proporción de los artículos del CG encontrados en el total de los resultados obtenidos por cada cadena en la base de datos. Como se puede ver en la Tabla 4.2 que la cadena con el ratio Y más alto es la cadena 3. Para asegurarnos que la cadena seleccionada sea la ideal para efectos de nuestra investigación, se calcula el promedio entre el ratio X y el ratio Y. De acuerdo con la Tabla 4.2, el promedio más alto es de la cadena 3. Esta cadena ostenta el mayor ratio Y (con un valor de 0.00548), además del promedio más alto (con un valor de 0.41941), por lo que es seleccionada como la cadena de búsqueda ganadora. Esta cadena se presenta en la Tabla 4.3.

Tabla 4.3: Cadena de Búsqueda Ganadora

Palabras Clave				
usability OR "user experience"	AND	evaluation OR testing	AND	tool OR tools OR software OR system AND automated OR automatic

Aunque Scopus es la base de datos que más resultados entrega (ver Tabla 4.4), también se consideran las base de datos IEEE Xplore y Web of Science para tener más resultados acorde a la temática de investigación del presente trabajo. En la búsqueda, solo se consideran estudios desde el 2016 hasta septiembre del 2021, mes en que se terminó de extraer estudios de las bases de datos. Los estudios duplicados en las bases de datos no se consideran, solo se mantiene el primero encontrado. En la Tabla 4.4 se muestran la cantidad de estudios encontrados en cada base de datos.

Tabla 4.4: Campos de Búsqueda Utilizados en cada Base de Datos

Base de Datos	Campos de Búsqueda	Total Resultados
Scopus	"Title OR Abstract OR Keywords"	904
IEEE Xplore	"Abstract"	162
Web of Science	"Title OR Abstract OR Keywords"	191

4.4. Criterios de Inclusión y Exclusión

Para la selección de estudios se definieron los siguientes criterios de inclusión:

a) El artículo describe una o varias herramientas que apoyen la evaluación de la usabilidad o experiencia de usuario, explicando con detalle su funcionamiento (algoritmos implementados, arquitectura, metodologías, teoría implicada, etc).

b) El artículo reporta una fase de pruebas en casos de uso reales donde se prueban las herramientas y se reportan resultados concluyentes, demostrando que la herramienta descrita cumple con el objetivo de apoyar la evaluación de la usabilidad.

Cabe destacar que para seleccionar un estudio, éste debe cumplir con ambos criterios de inclusión. En contraste a esto, se presentan los criterios de exclusión:

a) Las herramientas reportadas por el artículo no realizan, ni apoyan evaluación automática de la usabilidad.

b) El artículo no explica en detalle el funcionamiento de las herramientas presentadas.

c) El artículo no reporta una fase de pruebas de las herramientas.

d) La fase de pruebas reportada en el artículo no entrega resultados concluyentes que respondan a las PI.

e) Los resultados de la fase de pruebas reportada en el artículo no demuestran que las herramientas descritas cumplen con el objetivo de apoyar la evaluación de la usabilidad de forma automática.

f) Las herramientas descritas en el artículo entregan solo datos brutos, sin ningún tipo de análisis o crítica de estos.

g) La herramienta presentada en el artículo es un *framework*.

h) El artículo está escrito en un idioma diferente al inglés.

Cabe destacar que basta con que un estudio cumpla con uno de los criterios de exclusión para que sea descartado.

4.5. Selección de los Estudios Primarios

Usando la cadena de búsqueda ganadora en las bases de datos consideradas (Scopus, IEEE Xplore y Web of Science), se encontraron un total de 1811 estudios. Luego de eliminar los estudios duplicados entre las bases de datos, la cantidad de estudios se redujo a un total de 1257.

Se realizó una preselección de los estudios. Para esto se aplicaron los criterios de inclusión y exclusión al título y *Abstract* de cada uno de estos 1257 estudios. Los estudios se preseleccionaron siempre y cuando estuvieran en la misma línea de investigación presentada en este trabajo, es decir, que presentaran una herramienta que apoye la evaluación de la usabilidad de forma automática. Luego de realizar esta preselección, el total de estudios considerados se redujo a un total de 133 estudios. Este proceso fue realizado en gran parte por el estudiante, considerando que esto formaba parte de su trabajo de titulación. Se validaron dichos filtros de estudios con el profesor guía.

Luego de obtener estos 133 estudios, el paso siguiente fue aplicar de forma rigurosa los criterios de inclusión y exclusión explicados en la subsección anterior. Para esto, se descargaron los 133 estudios con el fin de ser leídos y analizados, determinando si estos cumplían con los criterios presentados y que además reportaran información que pudiera entrar en línea con la motivación de la investigación, además de buscar brindar respuesta a las PI.

Después de aplicar de forma rigurosa los criterios de inclusión y exclusión a todo el contenido de cada estudio, se redujo la cantidad de estudios a un total de 15 (ver Apéndice B). Con esto se termina el proceso de selección de estudios primarios. Los 15 estudios primarios serán los que permitirán cumplir con los objetivos de la investigación, responder a las PI y se podrá generar el cuerpo de conocimiento que englobe las herramientas que permiten apoyar la evaluación de la usabilidad de forma automática.

La Tabla 4.5 muestra un resumen de la cantidad de estudios según cada fase de la selección. Desde los primeros estudios encontrados tras el uso de la cadena de búsqueda en las bases de datos consideradas hasta la selección final de los estudios primarios, luego de aplicar rigurosamente los criterios de inclusión y exclusión.

La Figura 4.1 ilustra el proceso explicado en esta sección. Resumiendo, el proceso de selección se inicia aplicando la cadena de búsqueda en las tres bases de datos consideradas para esta investigación (Scopus, IEEE Xplore y Web of Science).

Tabla 4.5: Tabla Resumen

Base de Datos	Artículos Encontrados	Artículos sin Duplicados	Preseleccio-nados	Estudios Primarios
Scopus	912	904	110	13
IEEE Xplore	306	162	16	2
Web of Science	593	191	7	0
Total	1811	1257	133	15

Una vez se obtienen los artículos encontrados en las tres bases de datos (en total 1811), se eliminan los duplicados reduciendo el número total de artículos a 1257. Posteriormente, se aplican los criterios de inclusión y exclusión a este grupo de artículos, considerando sus títulos y *Abstracts*. Luego de una reunión de consenso, se preseleccionan 133 artículos. Después, se aplican los criterios de inclusión y exclusión de forma exhaustiva considerando el contenido de cada artículo, concluyendo con un total de 15 artículos seleccionados, artículos que satisfacen las PI definidas previamente.

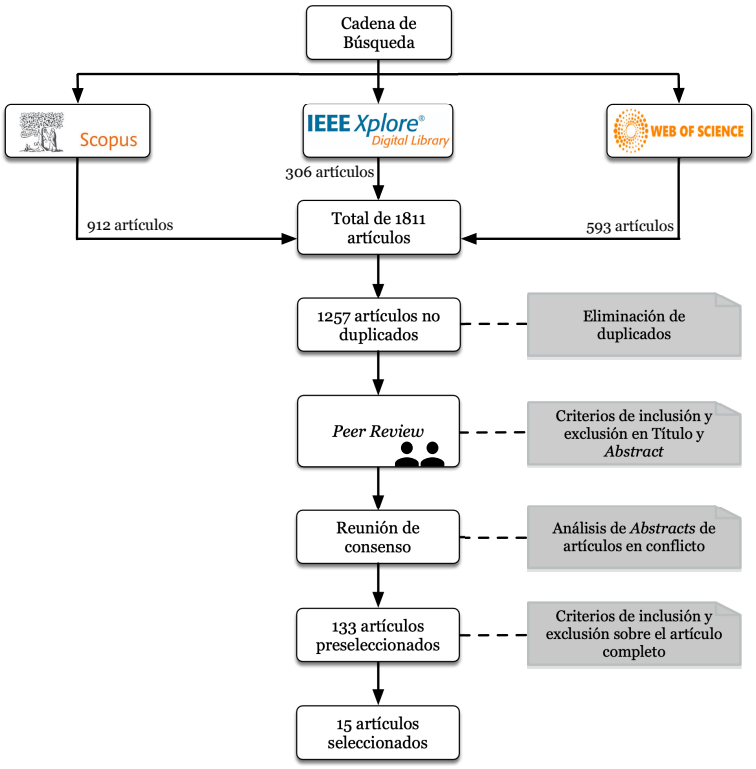


Figura 4.1: Diagrama de Selección de Papers

Capítulo 5

Resultados

De acuerdo con la metodología de investigación abordada en el capítulo anterior, se presentará en este capítulo los resultados conseguidos, junto con una síntesis de estos. Se buscará también dar respuesta a las PI. Considerando el ámbito de este trabajo y de acuerdo con la problemática principal que es la falta de un cuerpo de conocimiento que agrupe, clasifique y describa las herramientas que permiten apoyar la evaluación de la usabilidad de forma automatizada, se entregará dicho conocimiento utilizando los estudios primarios recabados.

La Figura 5.1 muestra la distribución por rango de fechas de los estudios primarios. En el año 2016 se encuentran solo dos estudios. Se puede apreciar un gran interés en herramientas que apoyan la evaluación de la usabilidad en el año 2017, donde se concentran cinco de los quince estudios primarios recabados. Este interés decae progresivamente, encontrándose tres estudios en el año 2018, dos estudios en el año 2019 y solo un estudio en el año 2020. Se recupera un poco el interés en esta materia de investigación en el año 2021, con dos estudios. Los estudios se agrupan de acuerdo al tipo de publicación, ya sea revista, capítulo de libro o conferencia.

En la Figura 5.1 se muestra también la clasificación de las herramientas que apoyan la evaluación de la usabilidad (se verá en detalle en la Sección 5.4). Se aprecia claramente que la clasificación que más herramientas engloba es “Herramientas que detectan problemas de usabilidad”, seguida de “Herramientas que apoyan la evaluación de la usabilidad”. Las que menos estudios engloban son “Herramientas que miden la usabilidad” y “Herramientas que corrigen problemas de usabilidad”.

Explicado lo anterior y presentados a grandes rasgos los resultados de la investigación, a continuación se responderán las PI generadas al principio de este trabajo. Cada una de estas cuatro preguntas serán respondidas en las siguientes secciones.

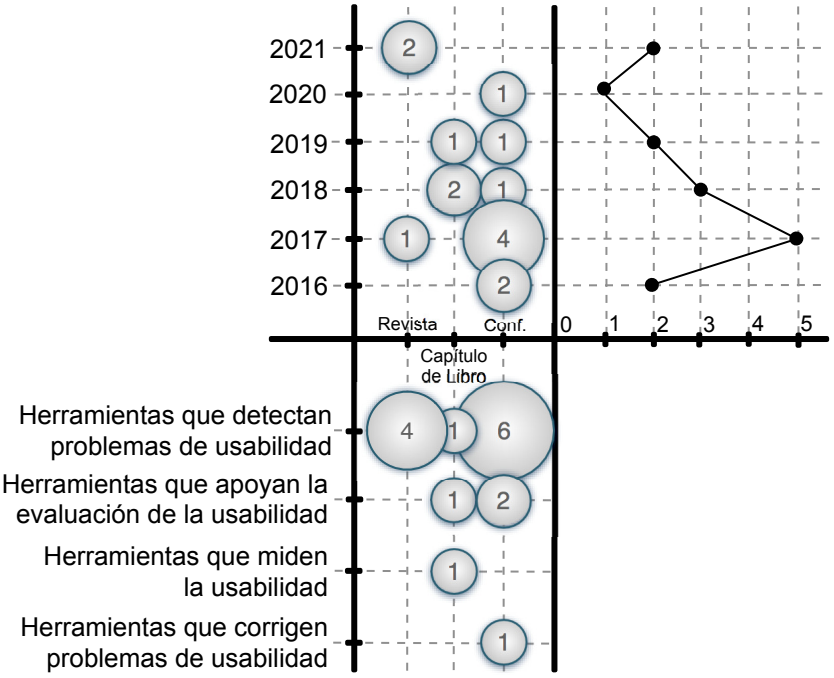


Figura 5.1: Diagrama de la Distribución de los Estudios Primarios

5.1. Herramientas Automatizadas para Apoyar la Evaluación de la Usabilidad

En esta sección se busca responder a la primera pregunta de investigación: (PI1) ¿Cuáles son las herramientas automatizadas que apoyan la evaluación de la usabilidad? Para responder a esta pregunta, se utilizó el total de 15 estudios primarios identificados en el SMS, los cuales presentan herramientas que apoyan la evaluación automatizada de la usabilidad. De estos 15 estudios se obtiene un total de 14 herramientas que cumplen con dicha función.

A continuación, se presentará el listado de las herramientas, una breve descripción de las mismas y la referencia correspondiente. En el Apéndice E se explicará con detalle cada una de las herramientas, presentando su funcionamiento y detalles técnicos, de acuerdo con lo reportado en los estudios primarios.

a) **MOBILICS** [Gonçalves *et al.*, 2016] (ver sección E.1)

MOBILICS es una extensión de USABILICS, por lo que hereda su metodología. Esta extensión surge de la necesidad de evaluar la usabilidad de páginas web en entornos móviles. Al existir la herramienta USABILICS, se realizaron las extensiones de sus actividades considerando los elementos *touch* propios de dispositivos móviles. Esta herramienta realiza la evaluación de la usabilidad comparando la interacción real del uso de un usuario realizando un test de usabilidad con la interacción predefinida por el evaluador que diseña el test.

b) ***Environment for Supporting Interactive Systems Evaluation*** [Assila *et al.*, 2016] (ver sección E.2)

Environment for Supporting Interactive Systems Evaluation es una herramienta que permite apoyar la evaluación de la usabilidad de forma automática de interfaces de usuario web de escritorio. Esta herramienta realiza la evaluación de la usabilidad detectando problemas de usabilidad mediante indicadores, utilizando datos de usabilidad obtenidos de métodos objetivos y subjetivos, utilizando cuatro herramientas para lograr esta integración. Las tres primeras son (i) *IESEval*, una herramienta que se ocupa de la evaluación de la interacción entre los usuarios y el sistema evaluado, obteniendo los datos de los eventos realizados en tests de usabilidad, (ii) una herramienta generadora de cuestionarios que soporta una evaluación subjetiva para evaluar las percepciones de los usuarios sobre el sistema evaluado permitiendo que las respuestas sean gestionadas y analizadas y (iii) un inspector de pautas ergonómicas que garantiza una evaluación de la usabilidad ergonómica de las interfaces de usuario. La cuarta, mencionada como herramienta de síntesis de los resultados de la evaluación, se encargada de integrar los resultados obtenidos por las tres primeras herramientas (objetivos y subjetivos) para poder detectar los problemas de usabilidad. Los datos de los cuestionarios realizados se relacionan con los datos obtenidos por *IESEval* y por la herramienta de inspección ergonómica de la usabilidad. Al final de los análisis de los resultados, la herramienta reporta los problemas encontrados con base en una serie de indicadores que pueden ser determinados por el evaluador. Estos indicadores muestran los elementos que están sujetos a problemas de usabilidad e indica recomendaciones.

c) **USF** (*Usability Smell Finder*) [Grigera *et al.*, 2017a] (ver sección E.3)

USF es una herramienta que permite apoyar la evaluación de la usabilidad de forma automática de interfaces de usuario web en entornos de escritorio, funcionando como Software-como-Servicio (SaaS o *Software-as-a-Service* por sus siglas en inglés). Esta herramienta realiza la evaluación de la usabilidad con un enfoque orientado a la detección de *usability smells*, los cuales sirven como pistas que apuntan a posibles problemas de usabilidad.

d) **MUSE** (*Mobile Usability Smell Evaluation*) [Paternò *et al.*, 2017] (ver sección E.4)

MUSE es una herramienta que permite apoyar la evaluación de la usabilidad de forma automática de interfaces de usuario web en entornos tanto de escritorio como móvil. MUSE registra la interacción de los usuarios en sesiones de pruebas de usabilidad. Gracias a su enfoque de servidor proxy, puede inyectar código JavaScript al sitio web que se desea evaluar sin la necesidad de que el propietario lo haga manualmente. MUSE permite al evaluador indicar las tareas a realizar en la aplicación a evaluar para que los usuarios realicen las pruebas de usabilidad. Las interacciones registradas por los usuarios se guardan con un indicador de tiempo, además de registrar los eventos de usabilidad con una captura de pantalla. La herramienta incluye un módulo de análisis que toma los datos de interacción, proporcionando información general de los registros recopilados. El análisis realizado por MUSE permite mostrar en la línea de tiempo los *bad usability smells* detectados, facilitando al evaluador la revisión de los eventos de usabilidad y los elementos afectados por estos problemas para su corrección.

e) **Kobold** [Grigera *et al.*, 2017b](ver sección E.5)

Kobold es una herramienta que permite apoyar la evaluación de la usabilidad de forma automática de interfaces de usuario web en entornos de escritorio, funcionando como SaaS (*Software-as-a-Service* por sus siglas en inglés). Esta herramienta realiza la evaluación de la usabilidad con un enfoque orientado a la detección de *usability smells*, entregando refactorizaciones que pueden ser implementadas de forma manual, semiautomática o automática para la corrección de problemas de usabilidad. Kobold está construida en base a USF, por lo que utiliza una estrategia similar a la hora de detectar *usability smells*. Consta de tres fases, las cuales son el registro de eventos de la interacción de usuario, la

detección de *usability smells* y la etapa de recomendaciones de refactorizaciones de usabilidad para solucionar los *usability smells* detectados.

f) **Plain** [Soui *et al.*, 2017](ver sección E.6)

Plain es una herramienta que permite apoyar la evaluación de la usabilidad de forma automática de aplicaciones móviles, por lo que su enfoque está en las interfaces de usuario móvil. Plain es un Plugin de Eclipse que permite predecir la usabilidad de una IU mediante la comparación de métricas de usabilidad con las propiedades de los elementos que conforman la IU móvil que se quiere evaluar. Estas métricas son de regularidad, composición, clasificación, complejidad (correspondientes a criterios de orientación), integridad, densidad, repartición y simetría (correspondientes a criterios de coherencia).

g) **UTAssistant** [Desolda *et al.*, 2017; Federici *et al.*, 2019, 2018](ver sección E.7)

UTAsisstant es una herramienta que permite apoyar la evaluación de la usabilidad de forma automática de interfaces de usuario con enfoque web. Se presenta como una plataforma web que permite apoyar la labor de la evaluación de la usabilidad brindando funciones que ayudan al evaluador con los datos registrados durante pruebas de usabilidad. UTAssistant permite recolectar los datos de registro de mouse y teclado durante las pruebas de usabilidad, además de permitir grabación de audio y video (tanto de pantalla como de la cara del usuario). Junto con esto, UTAssistant permite gestionar cuestionarios con el fin de almacenar los datos que los usuarios llenen para que esta pueda presentarlos en forma de estadística y gráficos.

h) **Guideliner** [Marenkov *et al.*, 2018](ver sección E.8)

Guideliner es una herramienta que permite apoyar la evaluación de la usabilidad de forma automática de interfaces de usuario web, tanto de entornos de escritorio como móviles. Guideliner se compone de varios módulos de Java y busca detectar problemas de usabilidad utilizando una amplia colección de pautas que permiten determinar estos problemas. Guideliner se presenta como una herramienta que, mediante una colección amplia de pautas, detecta problemas de usabilidad en interfaces de usuario web. Su motor de evaluación de usabilidad incorpora los elementos necesarios para poder realizar el proceso de forma automática. Guideliner utiliza *Selenium WebDriver* como motor de usabilidad, lo que permite

buscar y analizar los elementos de la IU y sus características. Con esto, se realiza la comparación de los valores obtenidos del análisis de la interfaz con los valores correspondientes a las pautas y se determina con esto los problemas de usabilidad.

i) **I2Evaluator** [Chettaoui y Bouhlel, 2017](ver sección E.9)

I2Evaluator es una herramienta que permite apoyar la evaluación de la usabilidad de forma automática de interfaces de usuario web, tanto de entornos móviles como de escritorio. I2evaluator está construida en base a un servicio web generado con AngularJS y busca medir las interfaces de usuario adaptables utilizando métricas estéticas (como por ejemplo, el balance de objetos de la IU, su densidad, la complejidad que presenta y su alineamiento), incorporando un algoritmo de descomposición de imagen. Para realizar el proceso de análisis de IU, I2Evaluator solicita al evaluador capturas de pantalla de la IU a evaluar. Cuando se proporcionan las capturas de pantalla, se solicita que se determine el tamaño de la ventana que se quiere expresar para determinar las métricas estéticas en base a este tamaño. Un algoritmo de descomposición de imagen ayuda a detectar los elementos de la IU para permitir que la herramienta realice los cálculos de las métricas.

j) **PlatoS** [Barra *et al.*, 2019](ver sección E.10)

PlatoS es una herramienta que permite apoyar la evaluación de la usabilidad de forma automática de interfaces de usuario en entornos de aplicaciones móviles. PlatoS permite la evaluación automática de *mockups*, detectando problemas de usabilidad en base al registro de la interacción del usuario con la interfaz a evaluar. El evaluador debe crear las tareas a realizar en las pruebas de usabilidad. Luego de esto, debe simular la interacción con la IU a evaluar de forma idónea. La herramienta PlatoS registra la secuencia de acciones y el tiempo en el que se ejecutan estas acciones. Las tareas y el *mockup* correspondiente son descargados por los usuarios finales para realizar las pruebas de usabilidad. Los usuarios realizan las tareas definidas por el evaluador y PlatoS registra la interacción y los tiempos de las acciones realizadas. Estas se comparan con la secuencia de acciones realizada por el evaluador. Utilizando métricas de usabilidad predefinidas, PlatoS realiza un análisis estadístico de los tiempos y las acciones realizadas por el

evaluador y los usuarios para detectar problemas de usabilidad, siendo estos informados al desarrollador del *mockup* para su corrección.

k) **OwlEye** [Liu *et al.*, 2020](ver sección E.11)

OwlEye es una herramienta que permite apoyar la evaluación de la usabilidad de forma automática de interfaces de usuario de aplicaciones móviles. Esta herramienta funciona implementando un modelo de CNN (*Convolutional Neural Network)* para la detección de problemas de usabilidad, mostrando los problemas de visualización de capturas de pantalla de interfaces de usuario de aplicaciones móviles. Se realiza un *data augmentation* basado en heurística para el entrenamiento del modelo CNN. Este se basa en un conjunto de datos de *Rico* que consta de 66.000 capturas de pantalla de más de 9.300 aplicaciones de Android. Estas se modifican mediante un algoritmo implementado por Liu *et al.* [2020] que permite tomar estas capturas de pantalla para crear pantallas con problemas de usabilidad. Con este conjunto y mediante el modelo de CNN, OwlEye puede detectar con un alto nivel de eficacia problemas en las interfaces de usuario que se quieran evaluar. En general, los problemas detectados se basan puramente en lo visual de la interfaz, por lo que estos errores pueden ser de oclusión de componentes, superposición de texto, imagen faltante, valores nulos y pantalla borrosa, entre otros. Destacar que esta herramienta simula la visualización de un experto de usabilidad a la hora de enfrentarse a inspeccionar una IU.

l) **ADUE** (*Automatic Domain Usability Evaluation*) [Bačíková *et al.*, 2021](ver sección E.12)

ADUE es una herramienta que permite apoyar la evaluación de la usabilidad de forma automática de aplicaciones de escritorio utilizando Java. ADUE detecta problemas de usabilidad de dominio basándose en el enfoque de usabilidad de dominio que abarca aspectos relacionados más con el contenido de los elementos que conforman la IU que con las características de los mismos. La información necesaria para realizar los análisis se extrae con DEAL (*Domain Extraction ALgorithm* por su siglas en inglés), que genera el modelo de dominio de la aplicación evaluada, además de filtrar los componentes estructurales que no contienen información de dominio. Con base en lo generado gracias a DEAL, la herramienta ADUE puede realizar un análisis ontológico, evaluar la especificidad, evaluar la gramática y realizar un análisis de *tooltip*. Con base en estas funcionalidades,

ADUE permite detectar problemas de usabilidad de dominio de la aplicación evaluada, mostrando al evaluador los errores y componentes asociados, además de entregar recomendaciones para poder corregir estos problemas.

m) **GTmetrix** [Al-Sakran y Alsudairi, 2021](ver sección E.13)

GTmetrix es una herramienta que permite apoyar la evaluación de la usabilidad de forma automática de páginas web en entornos de escritorio. Esta herramienta basa su funcionamiento analizando el rendimiento de los sitios web analizados considerando indicadores como la velocidad de carga de la página web y sus elementos. GTmetrix basa su funcionamiento en *Google's PageSpeed* y *Yahoo's YSlow*. Con esto, la herramienta es capaz de analizar el rendimiento de la página web. GTmetrix detecta problemas del rendimiento de las páginas web comparando las métricas de la página analizada con 23 reglas proporcionadas por Yahoo, las cuáles entregan valores promedio para aspectos del rendimiento de páginas web. Estos valores son comparados con los detectados en la página a analizar y se determina con esto los problemas a solucionar. GTmetrix también entrega informes de los análisis de rendimiento, entregando sugerencias de mejora y mostrando los puntajes obtenidos con base en las métricas de la página web analizada.

n) **Dareboost** [Al-Sakran y Alsudairi, 2021](ver sección E.14)

Dareboost es una herramienta que permite apoyar la evaluación de la usabilidad de forma automática de páginas web en entornos móviles. Esta herramienta utiliza métricas de rendimiento (por ejemplo, tiempos de carga, tamaño total de la página) para la evaluación de problemas de usabilidad, comparándolas con las métricas obtenidas de los elementos de la página web a analizar. A la hora de obtener estas métricas, Dareboost permite opciones para definir la geolocalización de donde se usa la página web para ver cómo se comporta en otros sectores geográficos, además de poder definir el tipo de navegador que se usa para el uso de la página web. También considera estas métricas de rendimiento de acuerdo con el sistema operativo móvil que se esté utilizando. La herramienta entrega informes donde se muestra el puntaje general de la página, el número de problemas y las mejoras recomendadas para sus respectivas correcciones.

Cada una de las herramientas descritas anteriormente, brinda un enfoque distinto a la hora de apoyar la evaluación de la usabilidad de forma automática. La mayoría de las herramientas presentadas permiten la detección de problemas de usabilidad.

5.2. Técnicas de Evaluación de la Usabilidad Beneficiadas por las Herramientas Automatizadas

En esta sección se busca responder a la segunda pregunta de investigación: (PI2) ¿Cuáles son las técnicas relacionadas con la evaluación de la usabilidad que se benefician de las herramientas automatizadas? Para esto, se mostrará cuáles fueron las técnicas específicas relacionadas con la evaluación de la usabilidad que fueron cubiertas por las herramientas que apoyan la evaluación de la usabilidad presentadas en los estudios primarios identificados con el SMS. Cabe destacar que estas herramientas tienen distintas funcionalidades y abarcan la evaluación de la usabilidad de manera diferente, por lo que las técnicas beneficiadas por el uso de éstas varían de acuerdo con el caso. Para interés del lector, en el Apéndice C se encuentra una tabla con todas las técnicas recolectadas por Ferré *et al.* [2002a,b].

Como resultado de esta investigación y de acuerdo con las herramientas presentadas en los estudios primarios, las técnicas de la evaluación de la usabilidad beneficiadas por el uso de estas herramientas, ordenadas de acuerdo con su índice de ocurrencia son:

a) Registro de la Interacción

b) Inspección de Conformidad con Estándares

c) Cuestionarios

d) Inspección de Consistencia

e) Revisión de Guías

f) Registro Continuo del Rendimiento del Usuario

g) Registro del Uso

h) Grabación de Video/Audio

i) Registro de Pulsaciones en el Tiempo

j) Métricas de Rendimiento

k) Evaluación Heurística

En las siguientes secciones, se presentarán las técnicas beneficiadas por el uso de herramientas automatizadas que apoyan la evaluación de la usabilidad, justificando cada herramienta en función de la técnica beneficiada.

5.2.1. Registro de la Interacción

Según Preece *et al.* [1994], el Registro de la Interacción es una técnica para la evaluación de la usabilidad que tiene como objetivo registrar la interacción completa de un usuario probando un sistema, de tal forma que pueda reproducirse completa en tiempo real. Esta se desprende del Registro Software, por lo que tiene como ventaja que no requiere que el investigador a cargo de la evaluación de la usabilidad esté presente, y no es obstrusivo.

Esta técnica fue la que más apareció a lo largo de la presente investigación (con un total de seis apariciones), considerando los estudios primarios recabados (ver Apéndice B). Las herramientas que apoyan esta técnica son las siguientes: *Environment for Supporting Interactive Systems Evaluation* [Assila *et al.*, 2016], USF [Grigera *et al.*, 2017a], MUSE [Paternò *et al.*, 2017], Kobold [Grigera *et al.*, 2017b], UTAssistant [Desolda *et al.*, 2017; Federici *et al.*, 2019, 2018] y PlatoS [Barra *et al.*, 2019].

La herramienta ***Environment for Supporting Interactive Systems Evaluation*** aborda la técnica de Registro de la Interacción gracias a uno de sus componentes principales, que es EISEval (*Environment for Interactive System Evaluation*). Recordar que de acuerdo con Assila *et al.* [2016], *Environment for Supporting Interactive Systems Evaluation* se basa en un entorno que incluye cuatro componentes: (i) EISEval, (ii)una herramienta que permite generar cuestionarios, (iii) un inspector de pautas de ergonomía y por último (iv) una herramienta que sintetiza los resultados de las tres herramientas anteriores y genera un análisis de usabilidad. El componente EISEval se basa en una evaluación objetiva y se enfoca en evaluar la interacción del usuario con el sistema evaluado, además de permitir capturar y analizar las acciones de los usuarios y sus interacciones con el sistema.

Para el caso de **USF**, Grigera *et al.* [2017a] explican que la arquitectura del proceso de registro de eventos se divide en un componente del lado del cliente que realiza el registro y un componente de lado del servidor que es responsable de la detección de *usability smells* y de la generación de informes. El componente del lado del cliente extrae eventos detallados para filtrar y agregar los relevantes, considerados como eventos de usabilidad. La implementación de esta herramienta en un caso de uso real de evaluación de la usabilidad permitió a los autores integrarla en una aplicación web, registrando los datos del interacción de usuario para encontrar los *usability smells*, que luego son informados al responsable de realizar la evaluación de la usabilidad.

MUSE es una herramienta basada en proxy que permite detectar problemas de usabilidad en páginas web [Paternò *et al.*, 2017]. La forma en la que esta herramienta aborda la técnica de Registro de la Interacción es gracias a su enfoque de detección de problemas de usabilidad. MUSE permite registrar el comportamiento de usuarios en pruebas de usabilidad, los datos obtenidos son usados para el análisis y detección de potenciales problemas de usabilidad. Los registros son guardados como eventos, guardando además su respectiva marca de tiempo de detección de evento y la etiqueta HTML donde se activó el evento, entre otros. Con estos registros, MUSE puede mostrar al evaluador en forma de *timeline* el comportamiento del usuario a lo largo de las pruebas de usabilidad, destacando los eventos realizados durante los tiempos correspondientes e indicando los potenciales problemas de usabilidad detectados con base en dichos registros.

La herramienta **Kobold** aborda la técnica de Registro de la Interacción de forma similar a USF, esto debido a que Kobold se desprende como trabajo futuro de USF, considerando que es un trabajo del mismo autor [Grigera *et al.*, 2017a,b]. Kobold se presenta como una mejora sustancial a la fórmula propuesta anteriormente por Grigera *et al.* [2017a]. Kobold realiza el proceso de registro de datos de interacción de los usuarios que prueban el sistema. Para implementarse, los que quieran realizar la evaluación de la usabilidad utilizando esta herramienta deben crear una cuenta para que esta proporcione un fragmento de código JavaScript que debe ser insertado en el código principal de la aplicación web. Con esto hecho, la herramienta comienza inmediatamente a registrar la interacción de los usuarios finales buscando, detectando y notificando los *usability smells*.

Para el caso de **UTAssistant**, Federici *et al.* [2018] explican que su herramienta permite diseñar y ejecutar tests de usabilidad para aplicaciones web (enfocándose en las plataformas web de la Administración Pública Italiana). Dentro de este diseño de test de usabilidad se debe tomar en cuenta la creación del script para introducir al usuario al test, definir el conjunto de tareas, identificar los datos que van a ser capturados (como el número de clics y el tiempo requerido por el usuario para completar una tarea, entre otros). Es en el apartado de implementación del test de usabilidad diseñado donde se puede apreciar el registro de la interacción, ya que una vez creado el test de usabilidad, éste es ejecutado para que el usuario forme parte de las pruebas. Con esto, la herramienta permite registrar las acciones del usuario en cada una de las tareas definidas previamente, recabando esa información y procesándola para entregar un análisis que permita a los evaluadores determinar los problemas de usabilidad.

Por último, en la herramienta **PlatoS** se aborda la técnica de Registro de la Interacción capturando los datos en archivos de registro que corresponden a la interacción del usuario con los *mockups* a evaluar. Recordar que la herramienta PlatoS, según lo explicado por Barra *et al.* [2019], busca evaluar la usabilidad de *mockups* o prototipos orientados a aplicaciones móviles mediante el registro y análisis de archivos de registro basados en la interacción de usuarios con estos *mockups*. En este caso, la información guardada en los archivos de registro corresponde a la interfaz que va a ser probada, la versión de la interfaz, la versión de PlatoS, sistema operativo y modelo del dispositivo, fecha y hora de simulación y datos de usuario.

A modo de síntesis, denotar que las herramientas que abordan la técnica de Registro de la Interacción tienen dentro de su funcionalidad la posibilidad de registrar acciones y datos del usuario en tests de usabilidad. Dependiendo de la herramienta, se registrarán distintos tipos y volúmenes de datos. Esta técnica es abordada de forma tal que las herramientas presentadas en esta sección pueden apoyar de forma automática en este aspecto y así, facilitar la implementación de una evaluación de la usabilidad para cualquiera que necesite registrar la interacción de un usuario con un sistema software específicamente.

5.2.2. Inspección de Conformidad con Estándares

Según Preece *et al.* [1994] y como su nombre indica, la Inspección de Conformidad con Estándares es un método de inspección donde especialistas de la tecnología

como los usuarios previstos, inspeccionan el sistema determinando si cumple con los estándares previamente propuestos. Constantine y Lockwood [1999] complementan esta definición agregando que el objetivo de ésta es identificar desviaciones de los estándares de IU o de las guías de estilo en vigor de la organización, por lo que todos los participantes de esta evaluación deben estar familiarizados con los estándares o guías de estilo aplicables.

Esta técnica de la evaluación de la usabilidad fue la segunda que más apareció a lo largo de este estudio (con un total de 4 ocurrencias). Las herramientas que apoyan esta técnica son las siguientes: Plain [Soui *et al.*, 2017], I2Evaluator [Chettaoui y Bouhlel, 2017], PlatoS [Barra *et al.*, 2019], y GTmetrix [Al-Sakran y Alsudairi, 2021].

Plain aborda la técnica Inspección de Conformidad con Estándares con base en su funcionamiento a la hora de apoyar la evaluación de la usabilidad. Plain es un *plug-in* de Eclipse que permite extraer las características de los elementos de las interfaces de usuario móviles basadas en Java. Soui *et al.* [2017] explican que usan un set de métricas de evaluación que fueron previamente validadas en estudios anteriores, métricas que se basan en Dirección (con métricas como la regularidad, composición, clasificación y complejidad) y Coherencia (con métricas como la integridad, densidad, repartición y simetría). Estas métricas son calculadas y comparadas con los valores extraídos de los elementos de la IU móvil evaluada. Si estos valores superan los umbrales establecidos por las métricas, se consideran problemas de usabilidad, asociando dicho problema al elemento de la interfaz analizado, determinando así si cumple con los estándares determinados.

En el caso de **I2Evaluator**, Chettaoui y Bouhlel [2017] explican que abordan la técnica de Inspección de Conformidad con Estándares con base en métricas de estética de interfaces de usuario de aplicaciones web implementada en un servicio web generado con AngularJS. I2Evaluator se basa en métricas de estándares de calidad como los de la ISO 25010 [ISO, 2011] (que aborda calidad de software/interfaces interactivas), además de métricas de balance y sencillez, las cuáles son calculadas por la misma herramienta. El análisis de estas métricas es llevado a cabo utilizando capturas de pantalla de la interfaz a evaluar. Un algoritmo de descomposición de imagen extrae los valores de los componentes detectados en la captura de pantalla y mediante la comparación de los valores de las métricas se pueden determinar los problemas de usabilidad, realizando así la inspección de conformidad.

PlatoS aborda la técnica de Inspección de Conformidad con Estándares mediante la funcionalidad de detección de problemas con actividades. PlatoS es una herramienta que permite apoyar la evaluación de la usabilidad de *mockups*. En la Sección 5.2.1 se menciona que PlatoS realiza el registro de la interacción mediante archivos de registro que guardan la interacción del usuario con el *mockup*. Con estos archivos de registro, se compara la interacción del usuario con métricas de usabilidad definidas por el diseñador mediante un análisis estadístico y se crea un informe detallado presentando dicho análisis.

Por último, en la herramienta **GTmetrix** se ve abordada la técnica de Inspección de Conformidad con Estándares. De acuerdo a lo explicado por Al-Sakran y Alsudairi [2021], GTmetrix permite evaluar la usabilidad de aplicaciones web de escritorio y está basada en Google's PageSpeed, Yahoo's YSlow e índices de rendimiento específicos. La inspección de conformidad con estándares se puede apreciar gracias a Yahoo's YSlow, ya que con esto la herramienta puede rastrear la plataforma web que se quiera evaluar y la compara con una lista de 23 reglas basadas en las reglas de Yahoo para sitios web de alto rendimiento, para luego calificar el sitio web según estas 23 reglas, brindando una calificación general basada en el promedio de estas.

A modo de síntesis, denotar que las herramientas que apoyan la técnica de Inspección de Conformidad con Estándares tienen dentro de su funcionalidad la comparación directa con métricas o reglas previamente definidas que, con base en un análisis de la herramienta, pueden determinar si cumplen con estos estándares y si son conformes con lo que se le exige a una IU de un sistema software en materia de calidad. Esta técnica es abordada de forma tal que las herramientas presentadas en esta sección pueden apoyar de forma automática en este aspecto y facilitar la implementación de una evaluación de la usabilidad para cualquiera que necesite garantizar la conformidad con estándares de calidad de un sistema software en específico.

5.2.3. Cuestionarios

De acuerdo con lo que propone Nielsen [1994], la técnica de Cuestionarios se trata de un método indirecto de estudio de la IU que permite conocer las opiniones del usuario sobre el uso de la UI, pero no información directa de ésta. Las preguntas y campos a rellenar por los usuarios de estos pueden ser definidos por los evaluadores o se pueden basar en estándares y guías para su confección. Son especialmente

apropiadas para obtener satisfacción subjetiva del usuario. Los cuestionarios pueden ser distribuidos por correo postal, correo electrónico o directamente con el software. Preece *et al.* [1994] complementa que hay dos tipos de preguntas: cerradas (las respuestas a las preguntas son un conjunto de alternativas y pueden estar basadas en una escala de valoración multipunto, en la escala Likert o en la escala de diferencial semántico) y abiertas (el encuestado puede dar libremente su propia respuesta).

Esta técnica de la evaluación de la usabilidad es apoyada por dos herramientas, según nuestra investigación. Estas herramientas son *Environment for Supporting Interactive Systems Evaluation* [Assila *et al.*, 2016] y UTAssistant [Desolda *et al.*, 2017; Federici *et al.*, 2019, 2018].

La herramienta propuesta por Assila *et al.* [2016], ***Environment for Supporting Interactive Systems Evaluation***, se presenta como un entorno que une cuatro componentes. Uno de éstos es una herramienta que genera cuestionarios de forma automática. Esta herramienta está enfocada en generar cuestionarios de usabilidad estandarizados o personalizados para asegurar una evaluación subjetiva de los sistemas interactivos a evaluar. Además, la herramienta permite capturar las percepciones de los usuarios sobre la usabilidad del sistema en general o sobre criterios de usabilidad más específicos. Con los datos rescatados de estos cuestionarios, se asegura una fase de análisis automática para ayudar a los evaluadores en la detección de problemas de usabilidad. De esta forma, la herramienta apoya la evaluación de la usabilidad brindando soporte a la técnica de Cuestionarios.

Por último, en el caso de **UTAssistant**, esta también permite apoyar la técnica Cuestionarios. Esta herramienta, según lo descrito por Federici *et al.* [2018] se enfoca en aplicaciones web. La herramienta facilita al usuario que diseña el test de usabilidad, la posibilidad de gestionar cuestionarios como parte del proceso de test de usabilidad. UTAssistant registra de forma automática las respuestas de los usuarios y expone los resultados en forma estadística y gráfica. Esto, junto con las otras funcionalidades de la herramienta, permiten apoyar y automatizar aspectos de la evaluación de la usabilidad.

A modo de síntesis, destacar que las herramientas que abordan la técnica de Cuestionarios tienen una funcionalidad que permite gestionar, generar y/o analizar cuestionarios para tests de usabilidad. Dependiendo de la herramienta, se gestionan las respuestas de estos de distinta forma, pero en general siempre dando un resultado y análisis de éstos (considerando presentar dichos resultados en forma de gráficos,

informes, estadísticas, comparaciones, etc.). Esta técnica es abordada de forma tal que las herramientas descritas y presentadas en esta sección permiten apoyar y automatizar aspectos de la evaluación de la usabilidad y la técnica Cuestionarios particularmente.

5.2.4. Inspección de Consistencia

Según Preece *et al.* [1994], la Inspección de Consistencia es una técnica relacionada con la evaluación de la usabilidad en la que un equipo de diseñadores se reúne con el fin de inspeccionar un conjunto de interfaces, para una familia de productos. Shneiderman [1997] complementa que los expertos verifican la consistencia a lo largo de una familia de interfaces, comprobando la consistencia de terminología, color, disposición, formatos de entrada/salida, etc. Por último, Constantine y Lockwood [1999] agregan a esta definición que el objetivo de esta inspección es identificar inconsistencias entre contextos de interacción y sus contenidos.

Esta técnica de la evaluación de la usabilidad es apoyada por dos herramientas, según nuestra investigación: *Environment for Supporting Interactive Systems Evaluation* [Assila *et al.*, 2016] y ADUE [Bačíková *et al.*, 2021].

Anteriormente se ha explicado la herramienta propuesta por Assila *et al.* [2016], ***Environment for Supporting Interactive Systems Evaluation***, la cual se presenta como un entorno que busca unir cuatro componentes. Entre estos componentes se encuentra un Inspector de Directrices Ergonómicas, el cual realiza una evaluación objetiva del sistema para evaluar la consistencia ergonómica de las interfaces de usuario. Este componente se basa en un conjunto de pautas ergonómicas, las cuales permiten detectar las inconsistencias y brindan una lista de recomendaciones para la IU que se está evaluando. Con esto, cumple la función de brindar retroalimentación al que realiza la evaluación de la usabilidad, apoyándose de la técnica de Inspección de Consistencia (en este caso con base en pautas ergonómicas) y utilizando esta herramienta como medio para hacerlo de forma automatizada.

Por último, según lo explicado por Bačíková *et al.* [2021] para la herramienta **ADUE**, esta permite la detección de problemas de usabilidad de dominio, un aspecto de la usabilidad enfocado más en el contenido de los elementos que conforman una IU que en sus características. Esta herramienta basa su funcionamiento completo en detectar problemas que amenacen con afectar la consistencia de la terminología y el contenido de los elementos de la IU. ADUE usa métricas de usabilidad de dominio

para medir esto, junto con realizar análisis ontológicos e inspecciones gramaticales. Mediante la detección de estos problemas, ADUE puede garantizar la consistencia del contenido de los elementos de la interfaz con base en su idioma, ortografía y correlación con los términos utilizados en toda la aplicación que se busca evaluar.

A modo de síntesis, destacar que las herramientas que abordan la técnica de Inspección de Consistencia brindan una funcionalidad que busca asegurar la consistencia con base en los elementos que conforman la IU. Dependiendo de la herramienta, estas usaran distintos métodos, como el uso de métricas o con algoritmos que aseguren la consistencia de los contenidos y elementos que conforman a las aplicaciones evaluadas. Esta técnica se aborda de forma tal que las herramientas descritas y presentadas en esta sección permiten apoyar la evaluación de la usabilidad y la técnica de Inspección de Consistencia, específicamente.

5.2.5. Revisión de Guías

Según Shneiderman [1997], la Revisión de Guías es una técnica de la evaluación de la usabilidad en la que se busca comprobar la conformidad de la IU con el documento de guías organizacional u otros. Ésta se desprende de la revisión por expertos, en la cual expertos en la aplicación o en el dominio de la UI realizan estas revisiones.

Esta técnica de evaluación de la usabilidad es apoyada por dos herramientas, según nuestra investigación: Guideliner [Marenkov *et al.*, 2018] y GTmetrix [Al-Sakran y Alsudairi, 2021].

Guideliner, herramienta presentada por Marenkov *et al.* [2018], permite apoyar la evaluación de la usabilidad de forma automática, considerando la conformidad de la IU de aplicaciones web. Esta herramienta basa su funcionamiento en el uso de guías de usabilidad, rescatando los elementos de la IU que se quiera evaluar y comparándola con estas guías. Guideliner también da la opción a la persona que quiera realizar la evaluación de la usabilidad de poder usar pautas definidas por él mismo. Las guías utilizadas se mencionan en el artículo que explica la herramienta Guideliner [Marenkov *et al.*, 2018]. De esta forma, la herramienta permite apoyar la evaluación de la usabilidad de forma automática utilizando la técnica de Revisión de Guías.

Por último, Al-Sakran y Alsudairi [2021] explican como **GTmetrix** apoya la evaluación de la usabilidad automatizada utilizando, entre otras técnicas, la Revisión de Guías. Como se ha explicado anteriormente, GTmetrix permite evaluar la usabili-

dad de aplicaciones web de escritorio y está basada en Google's PageSpeed, Yahoo's YSlow e índices de rendimiento específicos. Como se basa en Yahoo's YSlow, esta puede rastrear la plataforma web que se quiera evaluar y compararla con una lista de 23 reglas las que, además de comportarse como Estándares (ver Subsección 5.2.2), se pueden apreciar como guías que sirven para comparar los atributos de la IU que se está evaluando con información previamente validada que garantiza una correcta evaluación de la misma. Con esto, GTmetrix permite la evaluación automatizada de la usabilidad utilizando la técnica de Revisión de Guías.

A modo de síntesis, se destaca que las herramientas que apoyan la técnica de Revisión de Guías basan su funcionamiento en la guías o pautas que fueron previamente validadas como documentos organizacionales que dictan un cierto estándar. Estas herramientas brindan una función que permite realizar la comparación entre estas guías con el contenido y elementos de la IU que se quiera evaluar en materias de usabilidad. Las herramientas que fueron descritas y presentadas en esta sección permiten apoyar y automatizar aspectos de la evaluación de la usabilidad y la técnica de Revisión de Guías particularmente.

5.2.6. Registro Continuo del Rendimiento del Usuario

Según Shneiderman [1997], el Registro Continuo del Rendimiento del Usuario es una técnica de la evaluación de la usabilidad que se desprende de la evaluación durante el uso activo. En esta técnica, se destaca que la arquitectura software debería hacer fácil para los gestores del sistema recoger datos acerca de los patrones del uso del sistema, velocidad de rendimiento del usuario, tasa de errores o frecuencia de repeticiones de ayuda en línea.

Esta técnica para la evaluación de la usabilidad es apoyada por dos herramientas: *Environment for Supporting Interactive Systems Evaluation* [Assila *et al.*, 2016] y MUSE [Paternò *et al.*, 2017].

La herramienta propuesta por Assila *et al.* [2016], ***Environment for Supporting Interactive Systems Evaluation***, se presenta como un entorno que une cuatro componentes. Uno de estos es una herramienta, mencionada como IESEval, que refiere a las tareas de los usuarios en las que se utilizan las interfaces y todos los eventos generados por los dispositivos de interacción. Esta permite capturar un conjunto de medidas útiles para algunos criterios de usabilidad como la eficacia, la eficiencia y las acciones mínimas (explicadas en el artículo donde se presenta esta

herramienta [Assila *et al.*, 2016]). Estas medidas corresponden al rendimiento del usuario a la hora de realizar las pruebas de usabilidad, por lo que la herramienta presentada puede apoyar a esta técnica de la evaluación de la usabilidad.

Por último, la herramienta presentada por Paternò *et al.* [2017], **MUSE**, funciona con base en la técnica del Registro Continuo del Rendimiento del Usuario. MUSE es una herramienta que permite la evaluación automática de la usabilidad web basada en proxy que puede registrar el comportamiento de un usuario mientras interactúa con cualquier aplicación a través de dispositivos de escritorio o móvil. Para realizar esto, MUSE realiza la detección de problemas de usabilidad a través de un algoritmo de identificación de patrones de interacción específicos. Los datos del usuario se recopilan a través de código en JavaScript inyectado en la página web a través de un servidor proxy, por lo que MUSE puede comprobar las interacciones del usuario determinando el rendimiento que tuvo durante las pruebas de usabilidad, indicando así donde se encuentran los problemas de usabilidad. De esta forma, MUSE puede apoyar a esta técnica de la evaluación de la usabilidad.

A modo de síntesis, destacar que las herramientas que apoyan la técnica de Registro Continuo del Rendimiento del Usuario basan su funcionamiento en registrar la interacción del usuario determinando medidas que prueban el rendimiento de éste al realizar pruebas de usabilidad utilizando la interfaz que se quiere evaluar en materias de usabilidad. Estas herramientas brindan una función que permite recolectar la información del usuario y determinar estas medidas para encontrar problemas de usabilidad en el sistema software a evaluar. Esta técnica es abordada de forma tal que las herramientas presentadas en esta sección pueden apoyar de forma automática la evaluación de la usabilidad y la técnica de Registro Continuo del Rendimiento del Usuario específicamente.

5.2.7. Registro del Uso

De acuerdo con lo propuesto por Nielsen [1994], el Registro del Uso es una técnica para la evaluación de la usabilidad que busca registrar el uso real del usuario en su interacción con un sistema. Registrar el uso implica tener al ordenador recogiendo automáticamente estadísticas acerca del uso detallado del sistema. Normalmente es una forma de conseguir información acerca del uso de campo de un sistema tras su lanzamiento, pero puede utilizarse como un método suplementario en test de usabi-

lidad. Esta técnica es apoyada por una sola herramienta: MOBILICS [Gonçalves *et al.*, 2016].

La herramienta **MOBILICS**, explicada por Gonçalves *et al.* [2016], es una extensión de USABILICS que apoya de forma automática la usabilidad de interfaces de usuario en entornos web de dispositivos móviles. Para realizar la evaluación de la usabilidad automatizada, esta herramienta se basa en la detección de problemas a la hora de realizar tareas específicas en aplicaciones web. Los evaluadores generan una secuencia de uso de la IU con base en lo idóneo y la herramienta registra y compara el uso del usuario con esta interfaz. Con esta comparación de registro de uso, la herramienta puede detectar problemas de usabilidad. Al estar basado en USABILICS [Gonçalves *et al.*, 2016], MOBILICS busca incorporar eventos que se encuentran solo en los entornos de aplicaciones web para móviles, como pueden ser los eventos de interacción con pantalla *touch*. De esta forma, con el registro de uso del usuario comparándose con el uso de los evaluadores de usabilidad, se cubre de forma automatizada la técnica del mismo nombre.

En síntesis, toda herramienta que busque apoyar de forma automatizada la técnica de Registro del Uso debe tener dentro de sus funcionalidades la posibilidad de poder registrar el uso real del usuario en pruebas de usabilidad, con el fin de ocupar esta ruta de interacción con datos que reafirmen y que permiten comprobar donde se encuentran los problemas de usabilidad de la IU que se quiera evaluar. Con esto, la herramienta que se presente con estas características puede apoyar de forma automática la evaluación de la usabilidad y la técnica de Registro del Uso particularmente.

5.2.8. Grabación de Video/Audio

Hix y Hartson [1993] consideran la Grabación de Video/Audio como una técnica para la evaluación de la usabilidad que se desprende de las técnicas de recogida de datos, donde se busca obtener estos datos en tests de usabilidad o de cualquier tipo de observación de usuarios. Como su nombre bien indica, la Grabación de Video/Audio busca generar registros audiovisuales de la interacción de los usuarios con los sistemas en tests de usabilidad. Esta técnica de la evaluación de la usabilidad es apoyada por: UTAssistant [Desolda *et al.*, 2017; Federici *et al.*, 2019, 2018].

La herramienta **UTAssistant** permite dentro de sus funciones apoyar la evaluación de la usabilidad en entornos de aplicaciones web. Una de estas funciones

es permitir la grabación de video y audio a la hora de realizar tests de usabilidad. El evaluador que quiera ocupar UTAssistant puede diseñar evaluaciones de usabilidad y elegir qué tipo de datos quiere registrar. Cuando el evaluador envía el link al usuario para realizar la evaluación de la usabilidad, UTAssistant permite registrar estos datos de forma automatizada, grabando la voz del usuario por el micrófono, sus expresiones faciales mediante la webcam. Este contenido audiovisual permite a los evaluadores comprender el rendimiento en la ejecución de las tareas del usuario con la interfaz evaluada. Para apoyar a esto, UTAssistant proporciona herramientas de anotación, de modo que cuando los evaluadores detectan problemas de uso, pueden realizar estas anotaciones en las pistas de audio/video grabadas. Gracias a esta funcionalidad, se brinda apoyo a esta técnica para la evaluación de la usabilidad.

A modo de síntesis, las herramientas que buscan apoyar y automatizar la técnica de Grabación de Video/Audio deben tener dentro de sus funcionalidades la posibilidad de poder registrar de forma automática el audio (grabado generalmente por el micrófono del dispositivo donde se realiza los tests de usabilidad) y el video en el test de usabilidad (ya sea mediante la grabación directa de la pantalla del usuario o de su cara con cámaras mientras realiza los tests de usabilidad). Con esto, la herramienta que se presente con estas características puede apoyar de forma automática la evaluación de la usabilidad y la técnica de Grabación de Video/Audio específicamente.

5.2.9. Registro de Pulsaciones en el Tiempo

Según Preece *et al.* [1994], el Registro de Pulsaciones en el Tiempo es una técnica para la evaluación de la usabilidad que busca generar un registro de cada tecla pulsada por un usuario que prueba un sistema. Cada una de estas pulsaciones se almacena junco con el momento exacto en el que ha ocurrido el evento. Al desprenderse del registro software, cuenta con la ventaja de no requerir que un investigador a cargo de la evaluación de la usabilidad esté presente y no es obstrusivo. Esta técnica para la evaluación de la usabilidad es apoyada por una sola herramienta: PlatoS [Barra *et al.*, 2019].

La herramienta **PlatoS** [Barra *et al.*, 2019] es una herramienta que permite apoyar la evaluación automática de la usabilidad registrando la interacción del usuario con prototipos tempranos o *mockups* de aplicaciones móviles. Esta herramienta aborda específicamente la técnica de Registro de Pulsaciones en el Tiempo, ya que la

forma que tiene de registrar la interacción del usuario es mediante archivos de registro. Estos archivos de registro guardan, además de los datos de la interacción del usuario y la información de la interfaz a evaluar, los datos del componente de tiempo y fecha en que se realizan dichas interacciones. El responsable de realizar la evaluación de la usabilidad simula el uso del *mockup* con base en el uso ideal, haciendo que PlatoS registre la interacción considerando los tiempos en que se realiza cada una. Estos datos se transfieren al servidor de PlatoS y luego se compara la información de interacción del usuario que prueba la interfaz del *mockup* con los datos de interacción simulados por el evaluador mediante un análisis estadístico, creando un informe detallado de los resultados. Los tiempos de interacción son claves a la hora de determinar los problemas de usabilidad, es por esto que la herramienta PlatoS busca abordar esto con base en la técnica de Registro de Pulsaciones en el Tiempo.

A modo de síntesis, destacar que las herramientas que buscan apoyar y automatizar la técnica de Registro de Pulsaciones en el Tiempo deben tener dentro de sus funcionalidades la posibilidad de poder registrar de forma automática la interacción del usuario junto con el componente de tiempo de la realización de esta interacción. Como adicional, la herramienta debe permitir la comparación con base en otros datos de interacción con tiempo, para así poder determinar problemas de usabilidad. Con esto, la herramienta que se presente con estas características puede apoyar de forma automática la evaluación de la usabilidad y la técnica de Registro de Pulsaciones en el Tiempo particularmente.

5.2.10. Métricas de Rendimiento

De acuerdo con lo expuesto por Constantine y Lockwood [1999], Métricas de Rendimiento es una técnica para la evaluación de la usabilidad en la que se cuantifican importantes aspectos del uso real del sistema software a evaluar, ya sea en un entorno controlado de laboratorio o en el entorno habitual de trabajo. Esta técnica se desprende de las métricas de usabilidad, que ofrecen una forma de valorar la usabilidad del diseño de la IU que se está desarrollando. Esta técnica es apoyada por una sola herramienta: Dareboost [Al-Sakran y Alsudairi, 2021].

Dareboost se presenta como una herramienta que permite apoyar la evaluación automática de la usabilidad midiendo las métricas de rendimiento de interfaces de usuario de aplicaciones web para móviles [Al-Sakran y Alsudairi, 2021]. Dareboost mide el rendimiento del sitio web a analizar, indicando sus debilidades y destacando

advertencias, éxitos, tiempos de carga, tamaño total de la página y la cantidad de solicitudes HTTP, para luego presentar resultados por cada factor probado correspondiente a la IU que se quiera evaluar. Dareboost presenta los resultados de estas métricas de rendimiento con una puntuación general de la página, una cantidad de problemas, mejoras y éxito para la usabilidad del sitio web (cabe destacar que Dareboost también permite evaluar aspectos de la accesibilidad, pero eso escapa del alcance de nuestra investigación, ya que solo nos centramos en la evaluación de la usabilidad automatizada). Las métricas mencionadas son la base de funcionamiento de esta herramienta, siendo éstas usadas para los análisis correspondientes a la detección de problemas de usabilidad correspondientes a las interfaces de usuario web que se quieran evaluar.

A modo de síntesis, cabe mencionar que las herramientas que buscan apoyar y automatizar la técnica Métricas de Rendimiento deben tener dentro de sus funcionalidades la posibilidad de obtener las métricas que permitan determinar el rendimiento de una IU. Estas herramientas, consideran las características de los elementos de la IU y, en caso de evaluar la usabilidad de una aplicación web, buscan analizar los aspectos de conectividad que correspondan a éstas (como las solicitudes HTTP, la cantidad de texto que presenta la aplicación web, etc.). Además, las herramientas que aborden la técnica Métricas de Rendimiento deben permitir el análisis de los datos mencionados anteriormente y presentarlos al evaluador (ya sea mostrándolos directamente o en forma de nota o puntuación), de forma tal que permitan identificar problemas de usabilidad. Con esto, la herramienta que tenga estas características permitirá apoyar de forma automática la evaluación de la usabilidad y la técnica Métricas de Rendimiento específicamente.

5.2.11. Evaluación Heurística

Según Nielsen [1994], la Evaluación Heurística es una técnica para la evaluación de la usabilidad que se lleva a cabo observando una interfaz e intentando obtener una opinión acerca de lo bueno y malo. Para efectos prácticos, es bueno tener a varios evaluadores en materia de usabilidad que revisen el mismo diseño de forma independiente, puesto que con distintos puntos de vista se descubren muchos más errores que con un único evaluador. Es ideal que ésta sea realizada por especialistas en usabilidad. La técnica de Evaluación Heurística es apoyada por una sola herramienta: OwlEye [Liu *et al.*, 2020].

De acuerdo con lo explicado por Liu *et al.* [2020], **OwlEye** se presenta como una herramienta que permite apoyar la evaluación de la usabilidad de aplicaciones móviles. Esta evalúa la usabilidad utilizando un método de *data augmentation* basado en una heurística para generar capturas de pantalla de UI con problemas de visualización a partir de imágenes de interfaces de usuario que no tienen errores visuales, ni de usabilidad. Para esto, se necesita de un modelo de CNN para la comprensión visual, pero esto requiere de un gran volumen de imágenes para clasificarlas. Por lo tanto, Liu *et al.* [2020] exponen que se desarrolla un método que permite tomar estas capturas de pantalla a partir de imágenes de UI sin errores. Con esto, se busca simular el principal objetivo de la Evaluación Heurística como técnica para la evaluación de la usabilidad, que es llevar a cabo dicha evaluación por expertos en la materia que intenten obtener una opinión acerca de lo bueno y malo de una interfaz. Con este método de CNN se puede lograr (de acuerdo con los resultados expuestos por Liu *et al.* [2020]) realizar la evaluación de la usabilidad mediante la automatización de esta técnica.

A modo de síntesis, cabe mencionar que las herramientas que buscan apoyar y automatizar la técnica de Evaluación Heurística deben tener dentro de sus funcionalidades la posibilidad de poder realizar o emular una evaluación heurística, considerándola como el procedimiento en que expertos en materia de usabilidad inspeccionan la IU detectando sus puntos buenos y malos, concluyendo en posibles problemas de usabilidad. Cabe destacar que esta técnica tiene cierta dificultad en su implementación automatizada, ya que el concepto primordial de la evaluación heurística toma en cuenta el juicio subjetivo de los evaluadores expertos, una tarea complicada de parametrizar y automatizar. Con esto, la herramienta que se presente con estas características puede apoyar de una forma automática la evaluación de la usabilidad y la técnica de Evaluación Heurística particularmente.

5.3. Problemas y Retos con el Uso de las Herramientas Automatizadas

En esta sección se busca responder a la tercera pregunta de investigación: (PI3) ¿Cuáles son los problemas y retos existentes del uso de herramientas automatizadas para la evaluación de la usabilidad? Para esto, se obtiene la información de los estudios primarios que especifican y consideran ciertas dificultades y retos, ya sea en

el proceso de desarrollo de las herramientas que apoyan la evaluación de la usabilidad de forma automática, consideraciones a la hora de definir la arquitectura de estas herramientas, su marco teórico, etc.

De acuerdo con esto y considerando la información entregada por los autores de los estudios primarios, los principales problemas y retos identificados del análisis realizado son los siguientes:

a) Detección de eventos en sistemas software.

b) Detección de indicadores y umbrales.

c) Validación de métricas.

d) Se sigue necesitando de un experto de usabilidad.

e) Errores generales y mejora de rendimiento de herramientas.

f) Las herramientas no pueden reemplazar completamente la evaluación manual de la usabilidad.

A continuación, se describirán los problemas y retos mencionados, indicando el o los estudios primarios que abordan dicha problemática o reto explicando en qué consiste cada uno.

5.3.1. Detección de Eventos en Sistemas de Software

Esta problemática se basa en las dificultades reportadas por los autores referente a la detección misma de eventos en los sistemas software a evaluar utilizando las herramientas presentadas o desarrolladas. Se puede considerar esta dificultad o reto como un aspecto técnico del funcionamiento primordiale a la hora de utilizar la herramienta. Estas problemáticas, por lo general, son mostradas a la hora de realizar la fase experimental de la herramienta, la cual busca comprobar que tan bien funciona en materia de automatización de la evaluación de la usabilidad.

Gonçalves *et al.* [2016] exponen que, considerando la herramienta MOBILICS, el principal desafío a la hora de implementar estos eventos es cómo detectarlos correctamente. Destacar que ésta se basa en USABILICS, que principalmente se centra en evaluar de forma automatizada la usabilidad de entornos web de escritorio. Se propone con MOBILICS considerar los entornos web para dispositivos móviles,

por lo que se deben considerar nuevos eventos que no están presentes en sistemas web de escritorio. Considerando esto, se explica que los eventos básicos que desencadenan los oyentes son solo tres: *touchstart*, *touchmove* y *touchend*. Para solucionar esta problemática, los autores exponen que fue necesario reescribir de forma extensiva el código de JavaScript de la herramienta para poder detectar estos eventos.

5.3.2. Detección de Indicadores y Umbrales

Esta problemática se basa en las dificultades explicadas por los autores referente a la detección de indicadores y umbrales, generalmente relacionados con métricas, que permiten determinar mediante modelos de distinta índole los valores que deben tomar las características de elementos de la IU para no ser considerados como problemas de usabilidad. Este aspecto se puede considerar como una dificultad de implementación a la hora de determinar las métricas a utilizar y cómo capturar los datos que servirán para poder realizar las comparaciones correspondientes con los elementos de la IU a evaluar.

Assila *et al.* [2016] se refieren a esta problemática en el contexto de presentación de la herramienta *Environment for Supporting Interactive Systems Evaluation*. Recordar que el funcionamiento de esta herramienta, de acuerdo a lo abordado en la Sección 5.1, se basa en cuatro componentes, siendo la función de los tres primeros componentes el entregar datos que sirven para determinar problemas de usabilidad, los cuáles son tomados por el cuarto componente que realiza la síntesis de estos para entregar los resultados de la evaluación de la usabilidad automatizada. Ante esto, los autores destacan que son conscientes de que esta propuesta (refiriéndose a la herramienta propuesta) es un primer paso para sustentar las interpretaciones referentes a materias de usabilidad. Este tema es desafiante y requiere de más interés por parte de investigadores y profesionales en esta dirección, destacan Assila *et al.* [2016]. Ante esto, se puede desprender que para el 2016 la problemática giraba en torno a cómo interpretar estos umbrales e indicadores de forma tal que se tradujeran en valores y resultados certeros que, efectivamente, sirvieran para determinar problemas de usabilidad en interfaces de usuario.

5.3.3. Validación de Métricas

Esta problemática se basa en las dificultades explicadas por los autores referente al funcionamiento de las técnicas que requieren del uso de métricas y estándares de calidad, explicando que existe un cierto nivel de dificultad a la hora de elegirlas para que los resultados de detección de problemas de usabilidad sean certeros. En general, las herramientas que se basan en estos indicadores deben tener considerados modelos de análisis que permitan detectar aspectos de la IU y traducirlos a valores que puedan ser interpretados y comparados con estas métricas. Precisamente, este es el problema que reportan Chettaoui y Bouhlel [2017] que tiene la herramienta I2Evaluator. El enfoque que presentan los autores se refiere específicamente a la selección de métricas de acuerdo con las propiedades de las interfaces de usuario adaptables y su validez. Estas métricas son explicadas en detalle en el estudio primario [Chettaoui y Bouhlel, 2017]. Los autores señalan que es necesario comparar la métrica con la percepción del usuario y ajustarla hasta obtener un nivel adecuado de objetividad. Si bien se implementa este enfoque y tiene resultados de implementación que validan el funcionamiento de I2Evaluator, los autores afirman que se necesitan estudios empíricos más sólidos y generalizados para elegir cuidadosamente las métricas que cumplan de manera adecuada con cada una de las propiedades que se consideren oportunas de las interfaces de usuario adaptables. Esto se declara como trabajo futuro para la mejora de la herramienta I2Evaluator y los autores proponen una evaluación integral de la usabilidad de las interfaces de usuario adaptables.

5.3.4. Se Sigue Necesitando de un Experto de Usabilidad

Esta problemática se basa en que muchas herramientas no proveen de la información necesaria para poder determinar por sí solas problemas de usabilidad presentes en las interfaces de usuario que se quieran evaluar. Con esto, se puede decir que, a pesar que las herramientas presenten los datos recolectados de acuerdo con una extracción de características de elementos de la IU que se quiera evaluar, se necesitará de igual forma de un experto en materias de usabilidad que pueda sintetizar estos datos y concluir de estos los problemas de usabilidad correspondientes.

Grigera *et al.* [2017a] exponen en su artículo, referente a la herramienta USF, la existencia de estas herramientas. Éstas solo entregan datos duros sobre los elementos de la IU sin un previo análisis de éstos, señalando que para que estos datos

sean de alguna utilidad, un experto debe interpretarlos como problemas de usabilidad con base en su experiencia y criterio. Con esto, se señala que la idea que se debe tener de las herramientas que apoyan la evaluación de la usabilidad de forma automatizada es que, en su mayoría, deben brindar un análisis que permita que la misma herramienta detecte los problemas de usabilidad de una IU, que entregue retroalimentación y sugerencias para solucionar el error y, de ser posible, que corrija de forma automática dichos problemas. Señalar también que, en efecto, esta es la principal problemática que enfrentan las herramientas que buscan automatizar la evaluación de la usabilidad, ya que durante su desarrollo debe ser determinado de qué forma estas garantizarán resultados que permitan entregar una síntesis que defina los problemas de usabilidad de una IU. Evitar depender de expertos de la usabilidad es una de las problemáticas generales, considerando el contexto explicado. Con base en esta problemática es que Grigera *et al.* [2017a] desarrollan USF, considerando los puntos explicados.

5.3.5. Errores Generales y Mejoras de Rendimiento de las Herramientas

Esta problemática se desprende de lo mencionado por diferentes autores de los estudios primarios [Grigera *et al.*, 2017b; Marenkov *et al.*, 2018; Soui *et al.*, 2017] con base en la funcionalidad de las herramientas. Ninguna herramienta es perfecta, es por esto que los autores describen ciertos aspectos, limitaciones y consideraciones para las herramientas que desarrollan o exponen. Por lo general, estos son relatados y presentados como trabajos futuros, por lo que se desprende que dichos detalles están planificados para resolverse en versiones posteriores de dichas herramientas.

Grigera *et al.* [2017b] exponen en su artículo donde presenta la herramienta Kobold que considera como trabajos futuros comprobar la precisión en la detección de problemas de usabilidad para poder automatizar otras refactorizaciones que aún no se han implementado. Recordar que las refactorizaciones mencionadas por Grigera *et al.* [2017a] se refieren a mejoras incrementales de los problemas de usabilidad detectados, los cuales pueden ser detectados y notificados para su corrección o pueden ser corregidos por la herramienta Kobold, dependiendo del problema de usabilidad. Ante esto también exponen que otro desafío es seleccionar la refactorización más adecuada, sobre todo en los casos en que un problema de usabilidad o *usability smell* puede ser resuelto con diferentes refactorizaciones. Se destaca que todos estos

aspectos son mejoras que se pueden realizar de la herramienta en trabajos futuros, por lo que se pueden considerar como mejoras de su rendimiento.

Soui *et al.* [2017] señalan que los resultados del uso de la herramienta Plain indican que puede predecir de forma eficaz la usabilidad de las interfaces de usuario móvil, aunque explica que es necesario investigar algunos problemas, relacionados generalmente con la detección de los defectos de calidad que puedan ocurrir. Ante esto, planean algunas operaciones de refactorización como la reorganización del contenido de la interfaz del usuario móvil, el cambio de tamaño de los componentes detectados, etc. En general, se refieren a mejoras del rendimiento de la herramienta.

Marenkov *et al.* [2018] también destacan ciertos elementos a tener en cuenta con la herramienta Guideliner. Los autores señalan las limitaciones de la herramienta considerando que se enfoca en entornos web, indicando que páginas web que usen Flash o Java Applets no se consideran para el uso de Guideliner. Además, al estar basado en Selenium WebDriver solo soporta interfaces de usuario que utilicen como tecnologías HTML, JavaScript y CSS. Marenkov *et al.* [2018] también exponen que para que esta herramienta tenga una mayor distribución de uso es necesario que el código fuente de Guideliner sea refactorizado y se considere una colección mucho más amplia de guías y pautas (como las de *e-commerce*, bancos y motores de búsqueda). Todos estos aspectos son considerados como trabajos futuros para poder mejorar la herramienta.

En general, los autores que presentan el desarrollo de herramientas que evalúan la usabilidad de forma automatizada comentan que en trabajos futuros usarán sus herramientas en otros casos de uso para comprobar su eficacia, de lo cual se desprende que buscan mejorar su rendimiento.

5.3.6. Las Herramientas no Pueden Reemplazar Completamente la Evaluación Manual de la Usabilidad

Esta problemática se presenta debido a que, según los autores Al-Sakran y Alsudairi [2021], las herramientas automatizadas para la evaluación de la usabilidad no pueden reemplazar completamente aspectos de la evaluación de la usabilidad manual. Ante esto se destaca que hasta ahora todas las herramientas deben ser usadas por un usuario que se comporta como evaluador, el cuál usa los datos, recomendaciones y medidas que las herramientas entregan para corregir y destacar problemas de usabilidad.

Al-Sakran y Alsudairi [2021] explican la metodología que llevaron a cabo para probar las herramientas GTmetrix y Dareboost, teniendo como objetivo realizar una evaluación de la usabilidad de entornos web (de escritorio y móvil) apoyándose con estas dos herramientas. Señalan que para esto, realizaron una evaluación manual de la usabilidad y después una automatizada utilizando las herramientas mencionadas. Destacan que usar las herramientas considera varias ventajas, como la idoneidad para la evaluación a gran escala y un menor esfuerzo en términos de tiempo, pero se considera como esencial que las herramientas automatizadas no pueden reemplazar completamente las pruebas manuales, ya que para este caso, pueden encontrar problemas de usabilidad pero no demostrar cuán grave es dicho problema. Ante esto, es necesario disponer de un cierto criterio que debe ejercer el evaluador con base en la interpretación que quiera darle a la información entregada por la herramienta. Es de acá donde se desprende la problemática. Considerando el panorama general actual de las herramientas automatizadas de la evaluación de la usabilidad se puede afirmar que actualmente no pueden reemplazar aspectos de la evaluación de la usabilidad manual, por lo que se aconseja que se usen como un medio de apoyo de la evaluación más que como un reemplazo directo.

5.4. Clasificación de las Herramientas Automatizadas

En esta sección se busca responder a la cuarta y última pregunta de investigación: (PI4) ¿Cómo se pueden clasificar las herramientas automatizadas para la evaluación de la usabilidad? Para esto, se explicará la clasificación de las herramientas obtenidas luego de analizar los estudios primarios. Las clasificaciones reportadas buscan retratar el alcance de la herramienta explicada, considerando su funcionalidad y de qué forma apoya la evaluación de la usabilidad.

Un total de cuatro categorías fueron creadas de acuerdo con las herramientas reportadas en los estudios primarios. Las categorías son (i) Herramientas que miden la usabilidad, (ii) Herramientas que apoyan la evaluación de la usabilidad (iii) Herramientas que detectan problemas de usabilidad y (iv) Herramientas que realizan correcciones de problemas de usabilidad. A continuación, se explicará cada una de estas:

1. Herramientas que miden la usabilidad:

   Las herramientas que pertenecen a esta categoría realizan análisis de sistemas software y entregan un indicador (puede ser porcentaje de usabilidad, una calificación del 1 al 10, entre otros) que describe la usabilidad de dicho sistema. Estas herramientas no entregan un análisis muy detallado, ni tampoco entregan retroalimentación de los errores específicos del sistema analizado en materia de usabilidad. En esta categoría entra solamente la herramienta I2Evaluator [Chettaoui y Bouhlel, 2017].

2. Herramientas que apoyan la evaluación de la usabilidad:

   Las herramientas que pertenezcan a esta categoría realizan análisis de sistemas software y entregan un indicador que describe la usabilidad de un sistema y entregan funciones útiles que apoyan la labor de la evaluación de la usabilidad. Entre estas funciones adicionales se puede encontrar: (i) captura de datos automatizada (registro de eventos, archivos de registro, entre otros), (ii) generación de formularios para encuestas de usabilidad, (iii) *timelines* que apoyan la trazabilidad de la evaluación de la usabilidad, entre otros. En esta categoría entra solamente la herramienta UTAssistant [Desolda *et al.*, 2017; Federici *et al.*, 2019, 2018].

3. Herramientas que detectan problemas de usabilidad:

   Las herramientas que pertenecen a esta categoría realizan análisis de sistemas software y brindan retroalimentación de los errores específicos encontrados en materia de usabilidad. La herramienta mostrará estos errores en forma de alertas, informes, *warnings*, etc.

   A esta categoría pertenecen las herramientas MOBILICS [Gonçalves *et al.*, 2016], *Environment for Supporting Interactive Systems Evaluation* [Assila *et al.*, 2016], USF [Grigera *et al.*, 2017a], MUSE [Paternò *et al.*, 2017], Plain [Soui *et al.*, 2017], Guideliner [Marenkov *et al.*, 2018], PlatoS [Barra *et al.*, 2019], OwlEye [Liu *et al.*, 2020], ADUE [Bačíková *et al.*, 2021], GTmetrix [Al-Sakran y Alsudairi, 2021] y Dareboost [Al-Sakran y Alsudairi, 2021].

4. Herramientas que corrigen problemas de usabilidad

   Las herramientas que pertenecen a esta categoría realizan análisis de sistemas software y además de brindar retroalimentación de los errores específicos encon-

trados en materia de usabilidad, da la opción de corregirlos de forma automática. Se consideran en esta categoría herramientas que realicen la corrección de errores de forma totalmente automatizada (sin previa validación del usuario de la herramienta) o de forma semiautomática (usuario autoriza la corrección automática correspondiente). En esta categoría entra solamente la herramienta Kobold [Grigera *et al.*, 2017b].

Capítulo 6

Discusión y Amenazas a la Validez

En este capítulo se realiza una síntesis de los resultados obtenidos de la presente investigación. Las herramientas que apoyan la evaluación automatizada de la usabilidad son el principal resultado de esta, por lo que las PI giran en torno a ellas. El Capítulo está dividido en dos secciones. En la primera, se realiza la discusión generada por los resultados obtenidos y en la segunda se reportan las amenazas a la validez que pueden afectar a los resultados de la investigacón.

6.1. Discusión

Las herramientas identificadas y reportadas permiten apoyar la evaluación de la usabilidad de forma automática, por lo que se cumple con el objetivo general del presente trabajo de investigación. Con este cuerpo de conocimiento generado, esperamos que cualquier desarrollador o evaluador pueda usar las herramientas reportadas para facilitar la evaluación de la usabilidad.

El mapa mental presentado en la Figura 6.1 muestra un resumen de los cuatro aspectos fundamentales de la investigación realizada y de los resultados obtenidos por nuestro SMS: (i) Herramientas Automatizadas, (ii) Técnicas de Usabilidad Beneficiadas por las Herramientas, (iii) Problemas y Retos con el Uso de las Herramientas y (iv) Clasificación de las Herramientas Automatizadas. El centro de la Figura 6.1 corresponde al tema de investigación (Nivel 0). Cuatro ramas apuntan desde el centro, simbolizando los cuatro aspectos fundamentales mencionados anteriormente (Nivel 1). En las hojas del primer aspecto se listan las herramientas para la evaluación automática de la usabilidad, las del segundo aspecto presentan las técnicas bene-

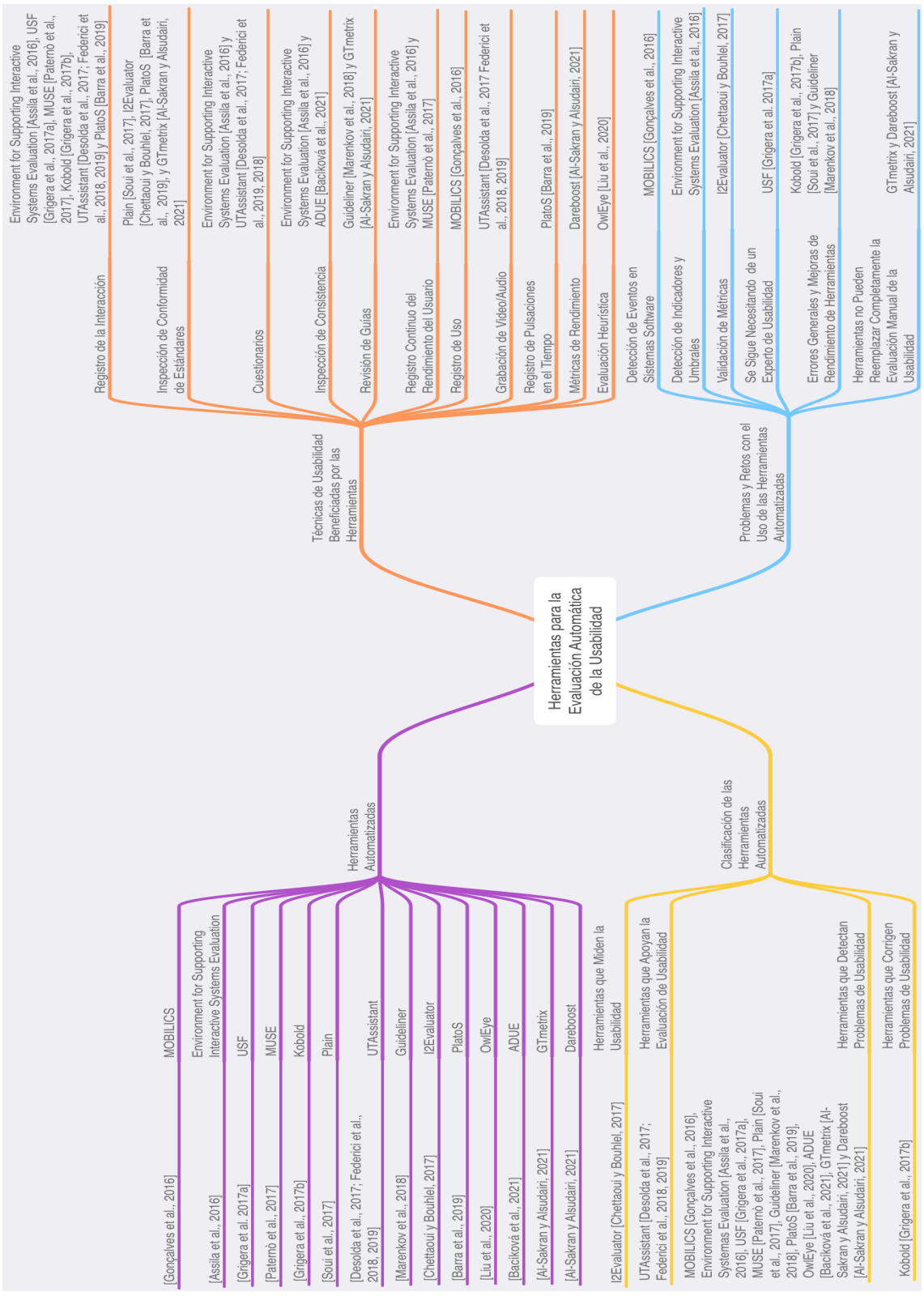


Figura 6.1: Aspectos generales sobre las herramientas que permiten apoyar la evaluación automática de la usabilidad.

ficiadas por las herramientas, las del tercer aspecto exponen los problemas y retos detectados y las del cuarto aspecto nombran las clasificaciones definidas para las herramientas (Nivel 2). En el Nivel 3 se relacionan las herramientas a cada una de las hojas del Nivel 2, según corresponda (exceptuando las hojas del primer aspecto). Estas son mostradas con la referencia correspondiente de los estudios primarios.

Debemos destacar la cantidad de estudios recolectados durante la investigación presentada en este trabajo. Luego de realizar la búsqueda en las bases de datos consideradas (Scopus, IEEE Xplore y Web of Science) se encontraron 1257 publicaciones (luego de eliminar los duplicados). Posterior a la aplicación de los criterios de inclusión y exclusión, el número de estudios resultante se redujo considerablemente, por lo que nos quedamos con un total de 15 estudios primarios. De entre estos estudios primarios, habían tres que hacían referencia a la misma herramienta (UTAssistant)[Desolda *et al.*, 2017; Federici *et al.*, 2019, 2018] y había un estudio que reportaba dos herramientas (GTmetrix y Dareboost) [Al-Sakran y Alsudairi, 2021]. Por lo que al final, el presente trabajo de investigación reporta un total de 15 herramientas que permiten apoyar la evaluación de la usabilidad de forma automática. A pesar de que fuimos exigentes con los criterios de inclusión y exclusión, debemos destacar que esperábamos una mayor cantidad de herramientas reportadas en la literatura (aún considerando que acotamos nuestra investigación a las tres bases de datos científicas mencionadas anteriormente). Como se explicó al inicio del Capítulo 5, el interés reflejado sobre las herramientas que permiten apoyar la evaluación de la usabilidad de forma automática se condensa en el año 2017, donde se destacan cinco estudios primarios. Se reduce el interés en esta temática entre los años 2018 al 2020, año en el que solo se encuentra un solo estudio primario. El número de estudios aumenta a dos en el año 2021. Esto puede significar que se esta retomando el interés en las herramientas que apoyan la evaluación de la usabilidad, pero se necesita más tiempo para determinar si sigue esta tendencia.

Con respecto a los resultados obtenidos con base en la clasificación de las herramientas identificadas en este trabajo de investigación, la clasificación con el mayor número de herramientas corresponde a aquellas que *detectan problemas de usabilidad*. Esto tiene sentido, ya que implementar mejoras de usabilidad en las aplicaciones evaluadas es muy complicado. Las herramientas que pertenecen a ésta categoría permiten generar recomendaciones a los desarrolladores para que estos puedan realizar las correcciones a los problemas detectados, algo que es de gran utilidad sobre todo

en etapas tempranas del proceso de desarrollo de software [Marenkov *et al.*, 2018]. La categoría *Herramientas que apoyan la evaluación de la usabilidad* destaca por presentar herramientas que, más que enfocarse en entregar evaluaciones y detección de problemas, buscan facilitar las tareas que conforman la evaluación de la usabilidad. Recolectar datos de cuestionarios, capturar la interacción del usuario, grabar audio y video a la hora de realizar tests de usabilidad son algunas de las funciones que presenta UTAssistant [Federici *et al.*, 2018], herramienta perteneciente a esta categoría. La categoría de *Herramientas que miden la usabilidad* incluye una sola herramienta, al igual que la categoría anterior, que es I2Evaluator. Esta herramienta presenta resultados sencillos, basándose en dar una puntuación con base en métricas que exponen la usabilidad de la IU evaluada (como por ejemplo, el balance de objetos de la IU, su densidad, la complejidad que presenta, entre otros). La categoría que más destaca es la de *Herramientas que corrigen problemas de usabilidad*. Esta categoría representa el idílico al que toda herramienta que permita la evaluación automática de la usabilidad debería apuntar. Una herramienta que permita detectar problemas de usabilidad y corregirlos de forma automática se presenta como una opción de alto valor si queremos seleccionar una herramienta que permita la evaluación automática de la usabilidad. La única herramienta perteneciente a esta categoría es Kobold [Grigera *et al.*, 2017b], la cual permite implementar refactorizaciones de forma automática y semiautomática de los elementos de las interfaces de usuario web cuando son detectados problemas de usabilidad potenciales. Si bien en la categoría de *Herramientas que detectan problemas de usabilidad* se encuentran herramientas de gran valor, no se puede negar el hecho de que una herramienta permita realizar correciones automáticamente se presenta como una más atractiva en materia de evaluación automática de la usabilidad.

Es importante destacar herramientas reportadas que resultan interesantes en materias de evaluación de la usabilidad automatizada. Una de estas herramientas, perteneciente a la categoría de *Herramientas que detectan problemas de usabilidad* es *Environment for Supporting Interactive Systems Evaluation* [Assila *et al.*, 2016], la cual tiene un enfoque único dentro de las herramientas identificadas en este trabajo de investigación, ya que detecta problemas de usabilidad integrando cuatro herramientas para realizar este proceso. Las tres herramientas primarias se encargan de obtener datos de usabilidad objetivos (capturando la interacción del usuario con la interfaz a evaluar y usando pautas ergonómicas para determinar la usabilidad de

la IU) y subjetivas (con una herramienta que gestiona cuestionarios para garantizar la opinión subjetiva del usuario). Usando una cuarta herramienta, esta permite sintetizar los resultados previos y detectar problemas de usabilidad.

Otra de las herramientas que destaca es MUSE, ya que brinda un enfoque simplificado y amigable al usuario a la hora de realizar la detección de problemas de usabilidad [Paternò *et al.*, 2017]. Esta herramienta registra la interacción del usuario al realizar tests de usabilidad. Al recoger esta información, MUSE genera una línea de tiempo con la secuencia de acciones de los usuarios que participaron de los tests, mostrando el tipo de interacción (o eventos) y las secuencia que presenta potenciales problemas de usabilidad. Esto permite a los desarrolladores comprobar de manera visual donde se ubican los errores de usabilidad en la IU gracias a la secuencia de acciones registrada.

UTAssistant es la herramienta con más estudios primarios que la respaldan, reflejando su importancia a la hora de implementar tests de usabilidad remotos para páginas web [Desolda *et al.*, 2017; Federici *et al.*, 2019, 2018]. Esta herramienta es la única reportada en este trabajo de investigación que presenta un enfoque más centrado en apoyar la evaluación de la usabilidad que en la detección de problemas. Esta herramienta facilita la gestión de datos que pueden ser de utilidad para los desarrolladores y evaluadores, registra la interacción del usuario, permite gestionar cuestionarios, graba audio y video (pantalla de usuario y expreciones faciales) y facilita la toma de notas a la hora de revisar estos datos. Resulta de gran utilidad disponer de una herramienta que permite generar estas soluciones, ya que al ejecutar tests remotos, es complicado disponer de datos como el lenguaje no verbal que ejerce una persona al realizar un test de usabilidad. Esta herramienta es la única que considera estos factores.

Guideliner [Marenkov *et al.*, 2018] se presenta como una opción sólida si lo que se quiere es disponer de un amplio abanico de guías y pautas validadas que garanticen la usabilidad de la IU que se está evaluando. Esta herramienta se basa en un conjunto de pautas para detectar problemas de usabilidad, una vez extraídas las propiedades de los elementos de la IU web y comparándolas con los valores establecidos por estas pautas. Si se quiere implementar una evaluación de la usabilidad completamente objetiva, esta herramienta logra su cometido. Además, brinda la opción de implementar pautas personalizadas, por lo que los desarrolladores y evaluadores tienen más libertad a la hora de elegir los parámetros que ellos consideren correctos.

No todas las evaluaciones de usabilidad deben estar regidas por los mismos marcos, por lo que la opción de permitir personalización de este proceso aporta bastante a la presentación de esta herramienta.

Una herramienta que destaca entre las que detectan problemas de usabilidad es OwlEye [Liu *et al.*, 2020], la cual ejecuta la técnica de evaluación heurística a la hora de realizar la evaluación de la usabilidad automática, utilizando CNN para simular la inspección visual de la IU como si lo estuviera haciendo un experto en materia de usabilidad. Este enfoque es único entre las herramientas identificadas en el presente trabajo de investigación. La herramienta I2Evaluator [Chettaoui y Bouhlel, 2017] usa algoritmos de descomposición de imágenes para detectar los elementos de la IU y determinar sus características comparándolas con métricas de usabilidad para medir la usabilidad de una IU. En cambio, OwlEye [Liu *et al.*, 2020] utiliza un set de miles de imágenes de interfaces de usuario con problemas de usabilidad para entrenar el modelo de CNN y poder determinar los problemas de usabilidad en las interfaces evaluadas con esta herramienta.

La evaluación de la usabilidad de interfaces de usuario se puede abordar de distintas formas, y es lo que exponen Bačíková *et al.* [2021] en su artículo reportando la herramienta ADUE. Esta herramienta se centra en evaluar la usabilidad de dominio de interfaces de usuario. Este concepto de usabilidad se centra más en el contenido lingüístico de los elementos que conforman la IU evaluada (ver Sección E.12). Si bien este trabajo de investigación no se centra en abordar los distintos enfoques de evaluación de la usabilidad, hay que destacar la metodología y enfoque único que Bačíková *et al.* [2021] presentan en su estudio.

De entre todas las herramientas reportadas, sin duda la más interesante es Kobold [Grigera *et al.*, 2017b]. Esta herramienta se basa en USF [Grigera *et al.*, 2017a], otra herramienta reportada en este trabajo de investigación y que es del mismo autor de Kobold. USF, por si sola, es una herramienta destacable para evaluar la usabilidad de sitios web de forma automática. Utilizando JavaScript es capaz de capturar la interacción del usuario con la interfaz evaluada y detectar de forma automática e inmediata potenciales problemas de usabilidad (definidos como *usability smells*). USF recomienda refactorizaciones para corregir estos problemas de usabilidad. Kobold toma esta metodología y permite que estas refactorizaciones puedan ser aplicadas a la IU web evaluada de forma automática o semiautomática cuando sea posible (ver Sección E.5). Si bien la herramienta Kobold no corrige automáticamente to-

dos los errores de usabilidad detectados, puede hacerlo con la mayoría. A pesar de lo explicado, Kobold no puede reemplazar totalmente a cualquier otro método de evaluación la de usabilidad. Esto se debe a que Kobold solicita información de refactorizaciones específicas al evaluador que utiliza la herramienta (refactorizaciones semiautomáticas). Con esto se refuerza la idea de que, aunque una herramienta pueda realizar correcciones automáticas a errores de usabilidad, siempre se necesitará de cierta implementación por parte de un evaluador o desarrollador.

Es importante destacar la contribución de este trabajo de investigación con respecto a los trabajos relacionados mencionados en el Capítulo 3 [Ivory y Hearst, 2001][Charfi *et al.*, 2014][Bakaev *et al.*, 2016][Khasnis *et al.*, 2019]. La investigación realizada se basa en un SMS para identificar estudios primarios que respondan a las PI, encontrando artículos que reportan herramientas que apoyan la evaluación automática de la usabilidad, considerando los criterios de inclusión y exclusión explicados en el Capítulo 4. El primer trabajo relacionado es el estudio de Ivory y Hearst [2001], en el que los autores centran sus esfuerzos en identificar aspectos de la evaluación de la usabilidad que se puedan automatizar, siendo precursores de investigaciones como la presentada en este trabajo. Además, los autores presentan herramientas que permiten apoyar la evaluación de usabilidad, pero estas no son tan sofisticadas como las que existen actualmente, razón por la cual en el presente trabajo se consideraron solo estudios publicados entre el 2016 y septiembre del 2021. El segundo trabajo relacionado fue el realizado por Charfi *et al.* [2014], en el cual los autores basan su investigación en *widgets* para la evaluación de la usabilidad, pero al igual que con el estudio presentado por Ivory y Hearst [2001], estos *widgets* no entran en el periodo de tiempo considerado para nuestra investigación, es decir, consideran *widgets* entre 1999 y el 2012, mientras que en nuestro estudio consideramos estudios más recientes. Además, destacar que en el estudio de Charfi *et al.* [2014] los autores no realizan un SMS. El tercer trabajo relacionado corresponde al realizado por Bakaev *et al.* [2016], en el cual los autores presentan herramientas para la evaluación de la usabilidad en sitios web solamente, mientras que nuestra investigación presenta un enfoque general en la búsqueda de herramientas para la evaluación automática de la usabilidad, no solamente para sitios web. Mencionar además que Bakaev *et al.* [2016] tampoco realizan un SMS. El cuarto trabajo relacionado fue el realizado por Khasnis *et al.* [2019], quienes enfocan sus esfuerzos en presentar herramientas para la evaluación de la usabilidad y, principalmente, relacionarla con las técnicas existentes para la

evaluación de la usabilidad, mientras que nuestra investigación se centra en explicar de forma exhaustiva las características de las herramientas recolectadas gracias a los estudios primarios seleccionados.

En general y con respecto a los trabajos relacionados, presentamos una investigación que permite conocer el panorama actual de las herramientas para la evaluación automática de la usabilidad, explicando dichas herramientas para brindar la mayor cantidad de información al lector que quiera implementar una evaluación de la usabilidad apoyándose en éstas. El enfoque que presentamos es distinto al de los trabajos relacionados y buscamos cubrir la necesidad de un cuerpo de conocimiento actual que encapsule las herramientas que permitan apoyar la evaluación de la usabilidad de forma automatizada, brindando una selección de herramientas reportadas en la literatura.

6.2. Amenazas a la Validez

La primera amenaza a la validez de nuestro trabajo de investigación es el sesgo en el proceso de selección de artículos (en la Sección 4.5 fue explicado este proceso). Destacar que la cantidad de estudios totales que obtuvimos luego de usar la cadena de búsqueda ganadora en las bases de datos consideradas fue de 1257 (luego de eliminar los duplicados). Los artículos encontrados con la cadena de búsqueda utilizada fueron evaluados de acuerdo con los criterios de inclusión y exclusión definidos en la Sección 4.4. Otros investigadores pueden haber evaluado las publicaciones de manera diferente. Para corroborar la concordancia en la selección de artículos se realizaron reuniones entre los investigadores para comprobar los artículos que fueron preseleccionados y los que se descartaron.

Otro punto relacionado con la selección de estudios primarios es el alcance declarado en nuestra investigación, ya que solo consideramos trabajos que fueron publicados en alguna de las bases de datos consideradas entre el 2016 y el 2021. Al considerar solamente este periodo de tiempo, podemos haber perdido algunos artículos relacionados directamente con nuestra investigación. Junto con esto, también se destaca que solo consideramos artículos científicos que estuvieran en inglés. Al descartar los demás idiomas, podríamos haber descartado algunos estudios que fueran relevantes para nuestra investigación. También podemos decir que en muchos estudios se hacían referencias a herramientas que podían ser interesantes, pero no cumplían con los

criterios de inclusión y exclusión, por lo que se podría haber realizado una búsqueda "bola de nieve", con el fin de encontrar más estudios.

Para el SMS realizado, consideramos solamente las bases de datos de artículos científicos Scopus, IEEE Xplore y Web of Science. Si bien encontramos una gran cantidad de resultados utilizando la cadena de búsqueda definida en la Sección 4.2, se podrían haber reportado más herramientas que apoyacen la evaluación de la usabilidad. Otro punto referente al alcance de nuestra investigación, es que no consideramos la literatura gris, la cual muy seguramente incluirá resultados que estén en línea con el objetivo del presente trabajo.

En nuestra investigación, se descartaron trabajos que reporten *frameworks* (ver Sección 4.4). La investigación se centra solamente en sistemas software reportados, pero incluir *frameworks* podría haber mostrado más herramientas interesantes para efectos de nuestra investigación.

La restricción enmarcada en la Sección 4.4 relacionada con seleccionar artículos de calidad que reporten una buena herramienta para la evaluación automatizada de la usabilidad implica que muchos artículos que presentaban herramientas fueron descartados por no responder las PI. Un criterio más flexible podría aumentar los resultados obtenidos.

Todas estas amenazas a la validez detectadas pueden ser abordadas en trabajos futuros relacionados con esta investigación, que busca reportar el panorama general de las herramientas que permiten apoyar la evaluación de la usabilidad de forma automatizada. Probablemente, existen más herramientas que permiten esto, y buscarlas de acuerdo a nuevos criterios y alcances puede beneficiar al panorama actual.

Capítulo 7

Conclusiones

En este capítulo se reportan las conclusiones finales obtenidas una vez terminado el trabajo de investigación. Se entregará una conclusión de acuerdo con cada una de las PI formuladas en el presente trabajo.

- PI1: ¿Cuáles son las herramientas automatizadas que apoyan la evaluación de la usabilidad?

De acuerdo con el SMS realizado, se pudo conocer el panorama general de las herramientas que permiten apoyar la evaluación de la usabilidad de forma automática reportadas en la literatura. Entre los años 2016 al 2021 se encontraron 15 estudios, en los cuáles se identificaron 14 herramientas que apoyan la evaluación de la usabilidad de forma automática.

Las herramientas reportadas en la Sección 5.1 y explicadas en el Apéndice E, muestran distintos enfoques para apoyar la evaluación de la usabilidad. Hay que destacar que estas se presentan con distintas metodologías y formas en las que permiten apoyar la evaluación de la usabilidad automatizada. Esto genera un amplio abanico de opciones para quienes deseen apoyarse en herramientas para realizar una evaluación de la usabilidad. La variedad de herramientas reportadas abarcan aplicaciones de escritorio, móviles y web (enfocadas en escritorio y móvil) para ser evaluadas, por lo que cualquier sistema software puede ser sometido a una evaluación de la usabilidad siendo apoyada por estas herramientas.

- PI2: ¿Cuáles son las técnicas relacionadas con la evaluación de la usabilidad que se benefician de las herramientas automatizadas?

Con base en las herramientas reportadas en el presente trabajo de investigación, se identificaron las técnicas de evaluación la de usabilidad cubiertas. Éstas técnicas no fueron definidas de forma explícita en todos los estudios primarios, por lo que se realizó un análisis de cada herramienta para identificar qué técnica de la evaluación la usabilidad ha sido abordada de acuerdo con su funcionamiento y enfoque. Las técnicas identificadas se pueden consultar en la Sección 5.2.

La técnica más abordada es registro de la interacción. Tiene sentido ya que una de los enfoques más utilizados en las herramientas reportadas es realizar tests de usabilidad para registrar la interacción de los usuarios con las interfaces evaluadas. Se puede destacar como las herramientas enfocan la métodología que utilizan en función de las técnicas de evaluación de la usabilidad. Algunas de las técnicas reportadas por Ferré *et al.* [2002a,b] fueron usadas ampliamente (por ejemplo, registro de la interacción), así como también se observó que hubieron técnicas que no se abordaron (como por ejemplo, interacción constructiva, método de entrenamiento, evaluación remota instrumentada, entre otras). Esto destaca que aún hay trabajo por realizar en materia de desarrollo de herramientas que apoyen la evaluación automática de la usabilidad. Cubrir las técnicas que aún no han sido abordadas es motivo para incentivar el desarrollo de estas.

- PI3: ¿Cuáles son los problemas y retos existentes del uso de herramientas automatizadas para la evaluación de la usabilidad?

De acuerdo con lo expuesto por los autores de los estudios, se pueden destacar ciertos desafíos, limitaciones y problemas existentes. La mayoría de autores reportan en sus arťiculos, además la herramienta correspondiente, los problemas que conlleva la implementación de la herramienta, sus limitaciones y algunos aspectos que se deben considerar y que presentan como trabajos futuros para mejorar dichas herramientas.

Un aspecto a destacar entre los retos reportados por los autores es la detección de eventos de la IU. Esto tiene sentido debido a que es la parte más importante cuando se realizan pruebas de usabilidad. Los problemas en la detección de eventos en las pruebas de usabilidad puede ocasionar resultados erróneos, lo que se traduce en una mala usabilidad para la interfaz evaluada (ver Subsección 5.3.1). Un aspecto similar

es la detección de indicadores y umbrales. Estos deben ser definidos y validados para que las herramientas que se enfocan en estos aspectos, puedan entregar una correcta evaluación de la usabilidad (ver Subsección 5.3.2). Además, se destaca la validación de métricas. Los autores destacan que elegir un conjunto de métricas que permitan a una herramienta evaluar la usabilidad de forma automática conlleva un alto nivel de dificultad. Cuando estas métricas no están previamente validadas, se deben considerar complejos modelos que permitan a la herramienta realizar análisis para la detección de problemas de usabilidad.

Grigera *et al.* [2017a] y Al-Sakran y Alsudairi [2021] afirman que si bien las herramientas ayudan, en gran medida, a la labor de realizar una evaluación de la usabilidad automatizada, se sigue necesitando de expertos de usabilidad en algunos casos y estas herramientas no pueden reemplazar completamente la evaluación manual (ver Subsecciones 5.3.4 y 5.3.6). Las herramientas reportadas utilizan distintas metodologías y tecnologías para apoyar la evaluaciíon de la usabilidad, pero al final estos resultados deben ser revisados por, en algunos casos, evaluadores expertos. Esto conlleva a que no se pueda reemplazar directamente a los evaluadores. Lo que si se puede rescatar de este punto es que, por lo menos, las herramientas reducen la necesidad de experticia en los evaluadores, permitiendo a personas con distintos niveles de conocimiento en materias de usabilidad, implementar una evaluación de la usabilidad con buenos resultados.

- PI4: ¿Cómo se pueden clasificar las herramientas automatizadas para la evaluación de la usabilidad?

Según el análisis realizado a los estudios primarios, las herramientas se pueden clasificar de acuerdo con su enfoque y con las funcionalidades que permiten apoyar la evaluación de la usabilidad automatizada. La clasificación de las herramientas es: (i) Miden la usabilidad, (ii) Apoyan la evaluación de la usabilidad, (iii) Detectan problemas de usabilidad y (iv) Corrigen problemas de usabilidad (ver Sección 5.4).

La clasificación que más herramientas incluye corresponde a aquellas que detectan problemas de usabilidad. Esto se puede intuir debido a que es un alcance más amplio a la hora de enfrentarse a una evaluación de la usabilidad. Estas herramientas entregan recomendaciones que sirven como guía para que los desarrolladores de las aplicaciones evaluadas puedan corregir los errores de usabilidad detectados. Un alcance más complejo es que la herramienta, de forma automática, detecte problemas de usabilidad, por lo que es entendible que en nuestra investigación solo una

herramienta pertenezca a en esta categoría. Kobold se presenta como una de las herramientas más interesantes debido a que permite integrar refactorizaciones de forma automática y semiautomática de elementos de interfaces de usuario web (ver Sección E.5). La categoría de herramientas que apoyan la evaluación de la usabilidad presenta herramientas que destacan por realizar un análisis de las interfaces de usuario evaluadas, además de automatizar aspectos que facilitan la labor de los evaluadores (como por ejemplo, el registro de la interacción de usuarios en pruebas de usabilidad) a la hora de implementar una evaluación de la usabilidad. La categoría de herramientas que miden la usabilidad se presenta como una alternativa sencilla a la hora de enfrentarse a una evaluación de la usabilidad. Esta presenta una sola herramienta, I2Evaluator, y se destaca por entregar medidas de usabilidad con base en métricas definidas (ver Sección E.9).

En trabajos futuros consideraremos tomar en cuenta más bases de datos de artículos científicos, como ACM Digital Library, SpringerLink y ScienceDirect. Considerar estas, puede ampliar los resultados a la hora de buscar herramientas que apoyen la evaluación de la usabilidad de forma automática. Otro aspecto relacionado sería incluir la literatura gris en los criterios de investigación. También se puede incluir en trabajos futuros a los *frameworks*. Esto podría reportar resultados interesantes o incluso centrar un estudio en los *frameworks* que apoyen la evaluación de la usabilidad de forma automática.

Referencias

Al-Sakran, H. O. y Alsudairi, M. A. (2021). Usability and accessibility assessment of saudi arabia mobile e-government websites. *IEEE Access*, 9:48254–48275.

Assila, A., de Oliveira, K. M., y Ezzedine, H. (2016). An environment for integrating subjective and objective usability findings based on measures. En *2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS)*, pp. 1–12. IEEE.

Atlas.ti9 (2021). Atlas.ti 9 desktop trial (windows. `https://atlasti.com/`.

Bačíková, M., Porubän, J., Sulír, M., Chodarev, S., Steingartner, W., y Madeja, M. (2021). Domain usability evaluation. *Electronics*, 10(16):1963.

Bakaev, M., Mamysheva, T., y Gaedke, M. (2016). Current trends in automating usability evaluation of websites: Can you manage what you can't measure? En *2016 11th International Forum on Strategic Technology (IFOST)*, pp. 510–514. IEEE.

Barra, S., Francese, R., y Risi, M. (2019). Automating mockup-based usability testing on the mobile device. En Miani, R., Camargos, L., Zarpelão, B., Rosas, E., y Pasquini, R., editores, *Green, Pervasive, and Cloud Computing. GPC 2019*, volumen 11484 de *Lecture Notes in Computer Science*, pp. 128–143. Springer, Cham.

Charfi, S., Trabelsi, A., Ezzedine, H., y Kolski, C. (2014). Widgets dedicated to user interface evaluation. *International Journal of Human-Computer Interaction*, 30(5):408–421.

Chettaoui, N. y Bouhlel, M. S. (2017). I2Evaluator: An aesthetic metric-tool for evaluating the usability of adaptive user interfaces. En Hassanien, A. E., Shaalan,

K., Gaber, T., y Tolba, M. F., editores, *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics. AISI 2017*, volumen 639 de *Advances in Intelligent Systems and Computing*, pp. 374–383. Springer, Cham.

Constantine, L. L. y Lockwood, L. A. D. (1999). *Software for Use: A Practical Guide to the Models and Methods of Usage-centered Design*. ACM Press/Addison-Wesley Publishing Co., New York, USA.

Desolda, G., Gaudino, G., Lanzilotti, R., Federici, S., y Cocco, A. (2017). Utassistant: A web platform supporting usability testing in italian public administrations. En *DCPD@ CHItaly*, pp. 138–142.

Fabo, P. y Durikovic, R. (2012). Automated usability measurement of arbitrary desktop application with eyetracking. En *2012 16th International Conference on Information Visualisation*, pp. 625–629. IEEE.

Federici, S., Mele, M. L., Bracalenti, M., Buttafuoco, A., Lanzilotti, R., y Desolda, G. (2019). Bio-behavioral and self-report user experience evaluation of a usability assessment platform (utassistant). En *VISIGRAPP (2: HUCAPP)*, pp. 19–27.

Federici, S., Mele, M. L., Lanzilotti, R., Desolda, G., Bracalenti, M., Meloni, F., Gaudino, G., Cocco, A., y Amendola, M. (2018). UX evaluation design of UTAssistant: A new usability testing support tool for italian public administrations. En Kurosu, M., editor, *Human-Computer Interaction. Theories, Methods, and Human Issues. HCI 2018*, volumen 10901 de *Lecture Notes in Computer Science*, pp. 55–67. Springer, Cham.

Ferré, X. (2005). *Marco de integración de la usabilidad en el proceso de desarrollo software*. Tesis doctoral.

Ferré, X., Juristo, N., y Moreno, A. M. (2002a). *Deliverable D.5.1. Selection of the Software Process and the Usability Techniques for Consideration*. STATUS Project (code IST-2001-32298) financed by the European Commission from December of 2001 to December of 2004. Available at http://is.ls.fi.upm.es/status/results/STATUS_D5.1_v1.0.pdf.

Ferré, X., Juristo, N., y Moreno, A. M. (2002b). *Deliverable D.5.2. Specification of the Software Process with Integrated Usability Techniques*.

STATUS Project (code IST-2001-32298) financed by the European Commission from December of 2001 to December of 2004. Available at http://is.ls.fi.upm.es/status/results/STATUS_D5.2_v1.0.pdf.

Gonçalves, L. F., Vasconcelos, L. G., Munson, E. V., y Baldochi, L. A. (2016). Supporting adaptation of web applications to the mobile environment with automated usability evaluation. En *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, pp. 787–794.

Grigera, J., Garrido, A., Rivero, J. M., y Rossi, G. (2017a). Automatic detection of usability smells in web applications. *International Journal of Human-Computer Studies*, 97:129–148.

Grigera, J., Garrido, A., y Rossi, G. (2017b). Kobold: web usability as a service. En *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 990–995. IEEE.

Hix, D. y Hartson, H. R. (1993). *Developing User Interfaces: Ensuring Usability Through Product & Process*. John Wiley & Sons, Inc., New York, NY, USA.

ISO (2011). 9126-1:2001. software engineering – product quality – part 1: Quality model.

ISO (2018). 9241-11:2018. ergonomics of human-system interaction–part 11: Usability: Definitions and concepts.

ISO/IEC (2007). 15939:2007. systems and software engineering – meassurement process.

Ivory, M. Y. y Hearst, M. A. (2001). The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys (CSUR)*, 33(4):470–516.

Khasnis, Shubhangi S and Aditi, Akanksha and Samrakshini, RS and Namratha, M and others (2019). Analysis of automation in the field of usability evaluation. En *2019 1st International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE)*, pp. 85–91. IEEE.

Kitchenham, B. A., Budgen, D., y Brereton, O. P. (2011). Using mapping studies as the basis for further research–a participant-observer case study. *Information and Software Technology*, 53(6):638–651.

Liu, Z., Chen, C., Wang, J., Huang, Y., Hu, J., y Wang, Q. (2020). Owl eyes: Spotting ui display issues via visual understanding. En *2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 398–409. IEEE.

Liyanage, N. L. y Vidanage, K. (2016). Site-ability: A website usability measurement tool. En *2016 Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer)*, pp. 257–265. IEEE.

Losana, P., Castro, J. W., Ferre, X., Villalba-Mora, E., y Acuña, S. T. (2021). A systematic mapping study on integration proposals of the personas technique in agile methodologies. *Sensors*, 21(18):6298.

Marenkov, J., Robal, T., y Kalja, A. (2018). Guideliner: a tool to improve web ui development for better usability. En *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics*, pp. 1–9.

Mayhew, D. J. (1999). *The Usability Engineering Lifecycle: A Practitioner's Handbook for User Interface Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Nielsen, J. (1994). *Usability Engineering*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Paternò, F., Schiavone, A. G., y Conti, A. (2017). Customizable automatic detection of bad usability smells in mobile accessed web applications. En *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pp. 1–11.

Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., y Carey, T. (1994). *Human-Computer Interaction*. Addison Wesley, 1st edición.

Shneiderman, B. (1997). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 3rd edición.

Soui, M., Chouchane, M., Gasmi, I., y Mkaouer, M. W. (2017). Plain: Plugin for predicting the usability of mobile user interface. En *VISIGRAPP (1: GRAPP)*, pp. 127–136.

Zhang, H., Babar, M. A., y Tell, P. (2011). Identifying relevant studies in software engineering. *Information and Software Technology*, 53(6):625–637.

Apéndice A - Lista de Frecuencia de Palabras

En este apéndice se presenta una tabla listando las palabras obtenidas de acuerdo con la metodología explicada en la Sección 4.2. La Tabla A.1 consta de la palabra obtenida de los artículos del CG, seguida de su porcentaje de aparición, la frecuencia de ocurrencia y el peso asociado a cada palabra.

Tabla A.1: Lista de Frecuencia de Palabras Completa

Palabras	**Aparición (%)**	**Frecuencia**	**Peso**
Usability	100	1156	1
Evaluation	100	577	0.7496
User	100	388	0.6678
Tool	100	240	0.6038
Users	100	224	0.5969
Use	100	150	0.5649
Interface	100	147	0.5636
Application	100	121	0.5523
Testing	100	118	0.5510
Tools	100	114	0.5493
Interaction	100	98	0.5424
Software	100	89	0.5385
Interfaces	100	89	0.5385
Human	100	85	0.5368
Systems	100	82	0.5355
Applications	100	78	0.5337
Automatic	100	67	0.5290
Evaluating	100	64	0.5277
Study	100	61	0.5264
Evaluate	100	56	0.5242
Experts	100	56	0.5242
End	100	52	0.5225
Support	100	44	0.5190
Expert	100	32	0.5138
Guidelines	66.67	319	0.4713
Automated	83.33	105	0.4621
Measure	83.33	61	0.4431
Solution	83.33	56	0.4409
Computer	83.33	54	0.4400
Environment	83.33	53	0.4396
Automatically	83.33	36	0.4322

Continúa en la siguiente página

Tabla A.1 – *Continuación*

Palabras	**Aparición (%)**	**Frecuencia**	**Peso**
Techniques	83.33	28	0.4288
Experience	83.33	28	0.4288
Interactions	83.33	28	0.4288
Product	83.33	26	0.4279
Easy	83.33	22	0.4262
Studies	83.33	20	0.4253
Assess	83.33	17	0.4240
UI	50.00	323	0.3897
System	66.67	124	0.3870
Evaluated	66.67	51	0.3554
Tests	66.67	45	0.3528
Questionnaire	66.67	42	0.3515
Interact	66.67	28	0.3454
Assessing	66.67	28	0.3454
Assessment	66.67	26	0.3446
Technique	66.67	24	0.3437
Service	66.67	21	0.3424
Automating	66.67	20	0.3420
Supports	66.67	12	0.3385
User Experience	66.67	8	0.3368

Apéndice B - Estudios Primarios

En este apéndice se presenta la Tabla B.1 con el listado de los estudios primarios obtenidos de acuerdo con la metodología de investigación explicada en la Sección 4.5. La Tabla B.1 consta del ID asociado a cada estudio primario, su respectivo título y la referencia correspondiente.

Tabla B.1: Estudios Primarios

ID	Título	Referencia
1	Supporting Adaptation of Web Applications to the Mobile Environment with Automated Usability Evaluation	[Gonçalves *et al.*, 2016]
2	An Environment for Integrating Subjective and Objective Usability Findings based on Measures	[Assila *et al.*, 2016]
3	Automatic Detection of Usability Smells in Web Applications	[Grigera *et al.*, 2017a]
4	Customizable Automatic Detection of Bad Usability Smells in Mobile Accessed Web Applications	[Paternò *et al.*, 2017]
5	Kobold: Web Usability as a Service	[Grigera *et al.*, 2017b]
6	PLAIN: Plugin for predicting the Usability of Mobile User Interface	[Soui *et al.*, 2017]
7	UTAssistant: A Web Platform Supporting Usability Testing in Italian Public Administrations	[Desolda *et al.*, 2017]
8	UX Evaluation Design of UTAssistant: A New Usability Testing Support Tool for Italian Public Administrations	[Federici *et al.*, 2018]
9	Bio-Behavioral and Self-Report User Experience Evaluation of a Usability Assessment Platform (UTAssistant)	[Federici *et al.*, 2019]
10	Guideliner: A Tool to Improve Web UI Development for Better Usability	[Marenkov *et al.*, 2018]
11	I2Evaluator: An Aesthetic Metric-Tool for Evaluating the Usability of Adaptive User Interfaces	[Chettaoui y Bouhlel, 2017]
12	Automating Mockup-Based Usability Testing on the Mobile Device	[Barra *et al.*, 2019]
13	Owl Eyes: Spotting UI Display Issues via Visual Understanding	[Liu *et al.*, 2020]
14	Domain Usability Evaluation	[Bačíková *et al.*, 2021]
15	Usability and Accessibility Assessment of Saudi Arabia Mobile E-Government Websites	[Al-Sakran y Alsudairi, 2021]

Apéndice C - Catálogo de Técnicas para la Evaluación de la Usabilidad

En este apéndice se presenta la Tabla C.1 en la que se reportan las técnicas relacionadas con la evaluación de la usabilidad [Ferré *et al.*, 2002a]. Las herramientas para la evaluación automática de la usabilidad se basan en algunas de las técnicas de la Tabla C.1. Para cada tipo de actividad de la Ingeniería de Software (IS) se reporta el nombre genérico de la técnica, los nombres dados por los autores y la referencia correspondiente.

Tabla C.1: Técnicas IPO Relacionadas con Actividades de Evaluación en el Proceso de Desarrollo de Software (adaptada de Ferré *et al.* [2002a])

Tipo de actividad IS	**Nombre Genérico de la Técnica IPO**	**Nombre Dado por los Autores IPO**	**Referencia**
Evaluación por Expertos	Evaluación Heurística	Evaluación Heurística	[Hix y Hartson, 1993][Mayhew, 1999][Constantine y Lockwood, 1999][Nielsen, 1994][Preece *et al.*, 1994][Shneiderman, 1997]
	Inspecciones	Inspecciones de Conformidad de Estándares	[Constantine y Lockwood, 1999][Mayhew, 1999][Preece *et al.*, 1994]
		Revisión de Guías	[Mayhew, 1999][Shneiderman, 1997]
		Inspecciones de Consistencia	[Constantine y Lockwood, 1999][Mayhew, 1999][Preece *et al.*, 1994][Shneiderman, 1997]
		Inspecc. Usab. Colaborat.	[Constantine y Lockwood, 1999]
	Recorridos Cognitivos	Recorridos Cognitivos	[Constantine y Lockwood, 1999][Mayhew, 1999][Preece *et al.*, 1994][Shneiderman, 1997]

Continúa en la siguiente página

Tabla C.1 – *continuación*

Tipo de actividad IS	**Nombre Genérico de la Técnica IPO**	**Nombre Dado por los Autores IPO**	**Referencia**
Evaluación por Expertos	Recorrido Pluralístico	Recorrido Pluralístico	[Constantine y Lockwood, 1999][Mayhew, 1999][Nielsen, 1994][Preece *et al.*, 1994]
Test de Usabilidad	Pensar en Voz Alta	Toma de Protocolo Verbal Concurrente	[Hix y Hartson, 1993]
		Pensar en Voz Alta	[Constantine y Lockwood, 1999][Nielsen, 1994][Preece *et al.*, 1994]
		Test Formales de Usab. (en las etapas iniciales)	[Mayhew, 1999]
		Interacción Constructiva	[Nielsen, 1994]
		Test Retrospectivo	[Constantine y Lockwood, 1999][Hix y Hartson, 1993][Nielsen, 1994][Preece *et al.*, 1994]
		Toma de Incidentes Crit.	[Hix y Hartson, 1993]
		Método de Entrenamiento	[Nielsen, 1994]
	Medición del Rendimiento	Tareas de Referencia	[Preece *et al.*, 1994]
		Métricas de Rendimiento	[Constantine y Lockwood, 1999]

Continúa en la siguiente página

Tabla C.1 – *continuación*

Tipo de actividad IS	**Nombre Genérico de la Técnica IPO**	**Nombre Dado por los Autores IPO**	**Referencia**
Test de Usabilidad	Medición del Rendimiento	Test Formales de Usab. (en etapas avanzadas)	[Mayhew, 1999]
	Información Post-Test	Información Post-Test	[Constantine y Lockwood, 1999]
	Test de Usabilidad en Laboratorio	Test en Laboratorios	[Constantine y Lockwood, 1999][Hix y Hartson, 1993]
		Laboratorio de Usabil.	[Nielsen, 1994]
		Test de Usab. y Lab.	[Shneiderman, 1997]
	Test de Campo	Test de Campo	[Constantine y Lockwood, 1999][Hix y Hartson, 1993]
	Grabación Video	Grabación Video	[Hix y Hartson, 1993][Nielsen, 1994][Preece *et al.*, 1994]
	Grabación Audio	Grabación Audio	[Hix y Hartson, 1993]
		Protocolo Verbal	[Preece *et al.*, 1994]
	Registro del Uso	Instrumentación Interna de la Interfaz	[Hix y Hartson, 1993]
		Registro del Uso	[Nielsen, 1994]
		Registro Software	[Preece *et al.*, 1994]
		Registro Continuo del Rendimiento del Usuario	[Shneiderman, 1997]

Continúa en la siguiente página

Tabla C.1 – *continuación*

Tipo de actividad IS	**Nombre Genérico de la Técnica IPO**	**Nombre Dado por los Autores IPO**	**Referencia**
Test de Usabilidad	Registro del Uso	Registro de Pulsaciones en el Tiempo	[Preece *et al.*, 1994]
		Registro de la Interacción	[Preece *et al.*, 1994]
	Eval. por Control Remoto	Eval. por Control Remoto	[Mayhew, 1999]
	Test Remoto por Video-Conferencia	Test Remoto por Video-Conferencia	[Mayhew, 1999]
Estudios de Seguimiento de Sistemas Instalados	Observación Directa	Observación Directa	[Nielsen, 1994][Preece *et al.*, 1994]
		Observación Aleatoria	[Mayhew, 1999]
	Cuestionarios y Encuestas	Cuestionarios	[Nielsen, 1994]
		Cuestionarios y Encuestas	[Preece *et al.*, 1994]
		Encuestas	[Shneiderman, 1997]
	Entrevistas	Entrevistas	[Nielsen, 1994][Preece *et al.*, 1994][Shneiderman, 1997]
		Entrevistas Estructuradas	[Hix y Hartson, 1993][Preece *et al.*, 1994]
		Entrevistas Flexibles	[Preece *et al.*, 1994]
	Focus Groups	Focus Groups	[Nielsen, 1994][Shneiderman, 1997]
	Registro del Uso	Instrumentación Interna de la Interfaz	[Hix y Hartson, 1993]

Continúa en la siguiente página

Tabla C.1 – *continuación*

Tipo de actividad IS	**Nombre Genérico de la Técnica IPO**	**Nombre Dado por los Autores IPO**	**Referencia**
Estudios de Seguimiento de Sistemas Instalados	Registro del Uso	Registro del Uso Real	[Nielsen, 1994]
		Registros Software	[Preece *et al.*, 1994]
		Registro Continuo del Rendimiento del Usuario	[Shneiderman, 1997]
		Evaluación Remota Instrumentada	[Mayhew, 1999]
		Registros de Pulsaciones en el Tiempo	[Preece *et al.*, 1994]
		Registro de la Interacción	[Preece *et al.*, 1994]
		Monitores Software	[Mayhew, 1999]
	Retroalimentación del Usuario	Retroalimentación del Usuario	[Nielsen, 1994]
		Buzón de Sugerencias o Reporte de Errores en Línea	[Shneiderman, 1997]
		Servicio de Atención al Usuario en Línea	[Shneiderman, 1997]
		Foros	[Shneiderman, 1997]
		Revistas y Conferencias para Usuarios	[Shneiderman, 1997]
		Evaluación Remota Semi-Instrumentada	[Mayhew, 1999]

Apéndice D - Publicación Derivada

En este apéndice se presenta la publicación derivada de este trabajo de investigación, publicada en una conferencia especializada en el área de la Interacción Persona-Ordenador. La conferencia mencionada es la *24th International Conference on Human-Computer Interaction* (HCII'22).

Automated Tools for Usability Evaluation: A Systematic Mapping Study

John W. Castro[1(✉)], Ignacio Garnica[1], and Luis A. Rojas[2]

[1] Departamento de Ingeniería Informática y Ciencias de la Computación, Universidad de Atacama, Copiapó, Chile
john.castro@uda.cl, ignacio.garnica.14@alumnos.uda.cl

[2] Facultad de Ciencias Empresariales, Departamento de Ciencias de la Computación y Tecnologías de la Información, Universidad del Bío-Bío, Chillán, Chile

Abstract. Usability is one of the most critical indicators in determining the quality of a software product. It corresponds to how users can use a software system to achieve specific objectives with effectiveness, efficiency, and satisfaction. A usability evaluation is necessary to ensure that the software system is usable, but this has certain disadvantages (e.g., a high cost of time and budget for the evaluation to be implemented). While these disadvantages can be a bit daunting despite the benefits they provide, some tools can automatically generate and support usability testing. We conducted a systematic mapping study to identify the tools that support automatic usability evaluation. We identified a total of 15 primary studies. In addition, we classify the tools into four categories: measure usability, support usability evaluation, detect usability problems, and correct usability problems. We identified that the automatic evaluation of the usability of web platforms and mobile devices is the most interesting.

Keywords: Usability · Evaluation · Tool · Automation

1 Introduction

Currently, there is a growth of developed software systems, causing an increased demand for higher quality systems, which can be ensured with specific standardized measures and methods through different activities and techniques. One of the essential measures when developing a software system is usability [1]. Usability is the extent to which users use a system, product, or service effectively and with satisfaction, given a context of use [2]. Usability is also related to the acceptability, by users, of a specific system, considering that it is good enough to meet the needs of users [1]. To ensure that these requirements are met, the developed systems must undergo a usability evaluation [3, 4].

Despite the importance of usability evaluation for any software system, it has certain disadvantages, such as a high cost of time and budget given its characteristics. Additionally, some techniques related to usability evaluation need at least one usability expert to be implemented [3, 4]. Guidelines, metrics, and heuristics can guide this expert to

G. Meiselwitz (Ed.): HCII 2022, LNCS 13315, pp. 1–19, 2022.
https://doi.org/10.1007/978-3-031-05061-9_3

support the work of usability evaluation. However, this expert evaluator will always provide a certain level of subjectivity in their analysis [4, 5]. Although these disadvantages can be discouraging, despite the benefits they provide considering the finished software product, they can be mitigated by implementing usability evaluation tools [5–8].

Usability evaluation tools are systems that support this task. Many tools directly benefit usability evaluation activities in an automated way, allowing, for example, during a usability test to store user registration data such as (i) keystrokes, (ii) clicks made with the mouse, and (iii) the distances traveled by the mouse pointer, among others. These tools allow, in some cases, to analyze the data collected to provide feedback to developers and usability experts, providing information on usability errors and, depending on the tool, automatically correcting them [5–9].

Currently, there is a wide variety of these tools. However, the related literature is composed of a set of independent publications. To the best of our knowledge, no study has comprehensively focused on this literature nor reports on the current state of automated tools for usability evaluation. This research seeks to generate a body of knowledge to classify the tools that support the automatic evaluation of usability. For this, we conducted a systematic mapping study (SMS).

This paper is organized as follows. In Sect. 2, we present the related work. In Sect. 3, we describe the research method of the SMS. In Sect. 4, we discuss the results of the SMS. Section 5 presents possible threats to validity, and finally, the conclusions are presented in Sect. 6.

2 Related Work

From our pilot search, we found that there were only four [4, 10–12] literature reviews related to our research. The first paper by Ivory and Hearst [4] reported the state-of-the-art usability evaluation methods, organized according to a taxonomy that emphasizes the role of automation. Ivory and Hearst [4] focused their efforts on identifying aspects of usability evaluation automation that are useful in future research and suggested new ways to expand existing approaches to better support usability evaluation. This study is interpreted as a precursor to automated approaches that, over time, became the development of tools that allow automatic evaluation of usability. Throughout his study, several tools are named, although not as sophisticated as those that currently exist, considering the year of publication of this study.

The second paper [10] reported *widgets* to help testers in the early evaluation of user interfaces. The authors explain that these *widgets* can detect certain ergonomic inconsistencies in the design of user interfaces. This study does not perform an SMS, and it focuses on exposing the *widgets* that were known. The authors explain the *widgets* in terms of functionality and application and show their experimental phase where they are tested. This study shows the *widgets* in a period before the one we consider (i.e., between 2016 and 2021), so the study is not considered in our research work.

The third paper by Bakaev et al. [11] provided an overview of the methods and tools of traditional, semi-automated, and automated approaches to website usability evaluation. The main difference from our research work, apart from the fact that the authors do not perform an SMS, is that Bakaev et al. [11] focused only on tools that support automated

usability evaluation of web user interfaces. In contrast, we focus our efforts on knowing the current panorama of these tools, whether they are focused on the web and desktop applications and mobile devices. Furthermore, the work of Bakaev et al. [11] only briefly describe the tools.

Finally, Khasnis et al. [12] presented a series of tools that support the usability evaluation in their research work, briefly explaining their operation, advantages, and disadvantages. It is important to note that the authors do not perform an SMS, as in this study. Furthermore, Khasnis et al. [12] focused on relating automatic usability evaluation tools with usability evaluation methods. Our approach focuses on relating the reported tools to the catalog of usability evaluation techniques proposed by [13, 14].

After analyzing these papers, we find that the SMS reported in this paper differs from the above reviews in that it aims not only to identify the automated tools to support the usability evaluation but also to (i) identify the techniques related to evaluation that benefit from these tools, (ii) determine the existing problems and challenges of using automated tools for usability evaluation and (iii) classify these tools. None of the reviews in the literature address this issue. Therefore, it is necessary to investigate the current state of automated tools for usability evaluation.

3 Research Method

The secondary study presented in this paper has been developed following the guidelines established by Kitchenham et al. [15] for conducting an SMS. Following these guidelines, the activities we carried out were: (i) formulating the research questions, (ii) defining the search strategies, (iii) selecting the primary studies, (iv) extracting the data, and (v) synthesizing the extracted data.

3.1 Research Questions

The information extracted from the primary studies aims to answer the following research questions: (RQ1) What are the automated tools that support the usability evaluation? (RQ2) Which usability evaluation-related techniques benefit from automated tools? (RQ3) What are the existing problems and challenges of using automated tools for usability evaluation? (RQ4) How can automated tools for usability evaluation be classified?

3.2 Define the Search Strategy

The SMS begins with identifying the keywords, for which it is necessary to find an initial set of articles that answer the research questions. This set is known as the Control Group (CG). The CG is a set of research papers representing, as accurately as possible, the set of primary studies that answers the research questions of the SMS [16]. Furthermore, the CG serves as a source of training samples for refining search strings and determining the sensitivity of the search strategy defined for the SMS. Keep in mind that a highly sensitive search strategy will retrieve many results. However, many of these may be unwanted articles, and a more precise search strategy will retrieve a few articles. However, it may

miss many studies that may be useful for research. Therefore, the formation of a CG must have a balance between these two factors [16]. To form the CG, a traditional search for studies related to the research context and, according to the previous explanation, that responds to the research questions was carried out. As a result of this search, six studies were identified [5, 7, 8, 17–19]. Before building the search string, it is verified if the CG studies are found in the Scopus database since it is the one that hosts the most studies. Within Scopus, there are five of the six that belong to the CG; they are [5, 7, 8, 18, 19]. Therefore, we can ensure that Scopus is the best option for research.

To obtain the keywords, a table was generated with the frequency of all the words and combinations of words that appeared in the CG articles, with the help of the Atlas.ti 9 software [20]. We selected only those words directly related to the research questions and that were present in a significant percentage of the CG articles. Subsequently, each one of the words obtained was assigned a value from 0 to 1, determined by its frequency of use, so that the word most frequently repeated in the various CG articles had the value 1. Table 1 shows a fragment of the list of words obtained as a result of this selection process. It shows the words, the percentage of CG studies it appeared in (coverage), the frequency of its appearance throughout the CG studies, and its assigned weight, based on the two previous columns. The weight is calculated based on the percentage of appearance and the frequency as follows (see Eq. 1):

$$((Word\ \ coverage)/(Maximum\ \ coverage) + (Word\ \ frequency)/(Maximum\ \ frequency))/2 \quad (1)$$

Table 1. Fragment of the list of words obtained from the selection process.

Words	Coverage (%)	Frequency	Weight
Usability	100	1156	1
Evaluation	100	577	0.7496
User	100	388	0.6678
Tool	100	240	0.6038
Interface	100	147	0.5636

3.3 Formation of the Search String

Once the keywords were identified, several search strings were constructed. For constructing the strings, four components are considered that correspond to a classification of the words considered. To define the components, the context of this research was considered, that is, knowing the current panorama of automatic tools that allow the usability evaluation to be supported. The defined components were the following: (i) tools, (ii) automation, (iii) evaluation, and (iv) usability. The logical operator AND was used to join

each of these components, while the logical operator OR was used to include synonyms of words from the same component. A total of four search strings were constructed, as shown in Table 2. We used these strings to search for CG studies within the Scopus database. It is important to remember that five of the six CG studies are in the Scopus database.

Table 2. Search strings.

ID	Search string	Studies found	GC found	Ratio X	Ratio Y	Average
1	(usability OR "user experience") AND (evaluation OR testing OR measure OR evaluating OR study OR evaluate OR tests OR assess) AND (tool OR systems OR applications OR tools OR software OR system OR application OR product) AND (automated OR automatic OR automatically OR automating)	2620	5	0.8333	0.0019	0.4176
2	(usability) AND (evaluation OR testing OR measure) AND (tool OR systems OR applications) AND (automated OR automatic OR automatically)	1004	5	0.8333	0.0049	0.4191
3	(usability OR "user experience") AND (evaluation OR testing) AND (tool OR tools OR software OR systems) AND (automated OR automatic)	912	5	0.8333	0.0054	0.4194
4	usability AND (evaluation OR testing OR evaluate OR study) AND (tool OR software OR systems) AND (automated OR automatic)	1304	5	0.8333	0.0038	0.4185

Table 2 shows the number of studies found and the number of CG articles found for each search string tested. All search strings find all five GC studies. Because of this, it was necessary to use additional indicators. These indicators are the X ratio (see Eq. 2), the Y ratio (see Eq. 3), and the average between both (see Eq. 4).

$$XRatio = (No.\ of\ \ articles\ found\ \ in\ \ the\ \ control\ \ group) / (Total\ \ of\ \ articles\ \ in\ \ the\ \ control\ \ group) \quad (2)$$

$$YRatio = (No.\ of\ \ articles\ found\ \ from\ \ the\ \ control\ \ group) / (Total\ \ of\ \ articles\ found\ \ per\ \ search\ \ string) \quad (3)$$

$$Average = (XRatio + YRatio)/2 \quad (4)$$

As shown in Table 2, the X ratio remains the same for all search strings. This is because, with all strings tested in the Scopus database, the same number of articles belonging to the CG was found. However, the Y ratio shows specific differences since it is based on calculating the proportion of the CG articles found in the total of the results obtained by each string. The string with the highest Y ratio is string 3. To ensure that the selected string is the ideal one for our investigation, the average between the X ratio and the Y ratio is calculated. According to Table 2, string 3 has the highest average, so it is selected as the best search string. The structure of the final search string is shown in Table 3.

Table 3. Final search string.

Keywords						
usability OR "user experience"	AND	evaluation OR testing	AND	tool OR tools OR software OR system	AND	automated OR automatic

Although the search string tests were performed in Scopus, the largest database of peer-reviewed literature [21], the searches were also performed in the IEEE Xplore and Web of Science (WoS) in order to acquire more results. In the search, only studies from 2016 to September 2021 are considered. The databases were analyzed sequentially, using the search fields shown in Table 4. The search fields used were determined by the options provided by each database, due to the different query syntaxes [22–24]. If a duplicate appeared, the first result was kept.

Table 4. Search field per database.

Database	Search fields	Number of results
Scopus	"Title OR Abstract OR Keywords"	904
IEEE Xplore	"Abstract"	162
Web of Science	"Title OR Abstract OR Keywords"	191

3.4 Inclusion and Exclusion Criteria

The inclusion criteria used to select the primary studies are summarized below:

- The article describes one or several tools that support the evaluation of usability or user experience, explaining in detail its operation (e.g., implemented algorithms, architecture, methodologies, theory involved).
- The article reports a testing phase in actual use cases where the tools are tested, and conclusive results are reported, demonstrating that the described tool meets the objective of supporting the evaluation of usability.

It is essential to mention that selecting a study must meet both inclusion criteria. In contrast to this, the exclusion criteria are as follows:

- The tools reported in the study do not perform or support automatic usability evaluation.
- The article does not explain the operation of the presented tools in detail.
- The article does not report a testing phase of the tools.
- The testing phase reported in the article does not deliver conclusive results that answer the research questions.
- The results of the testing phase reported in the article do not show that the tools described meet the objective of supporting usability evaluation automatically.
- The tools described in the article deliver only raw data without any analysis or critique.
- The tool presented in the article is a framework.
- The article is written in a language other than English.

Note that it is enough for a study to meet one of the exclusion criteria to be discarded.

3.5 Select the Studies

A total of 1811 papers were found in the different databases. After excluding duplicate articles, the number was reduced to 1257. Subsequently, a selection of studies was made by applying the inclusion and exclusion criteria to the title and abstract of each of these 1257 studies. The selected articles were validated during a consensus meeting, in which we analyzed the abstracts of articles with conflicting decisions, thus reducing the total to 133 pre-selected articles. After the meeting, the selection criteria were again applied to the full text of the remaining articles. Figure 1 shows the entire filtering and analysis

process with the inclusion and exclusion criteria used to select 15 papers. A complete list of the primary studies can be found in Appendix A. The results of applying the different filters during the selection process for each database can be seen in Table 5.

Table 5. Number of remaining studies after filtering the database results.

Database	Studies found	Duplicate-free	Pre-selected studies	Primary studies
Scopus	912	904	110	13
IEEE Xplore	306	162	16	2
Web of Science	593	191	7	0
TOTAL	**1811**	**1257**	**133**	**15**

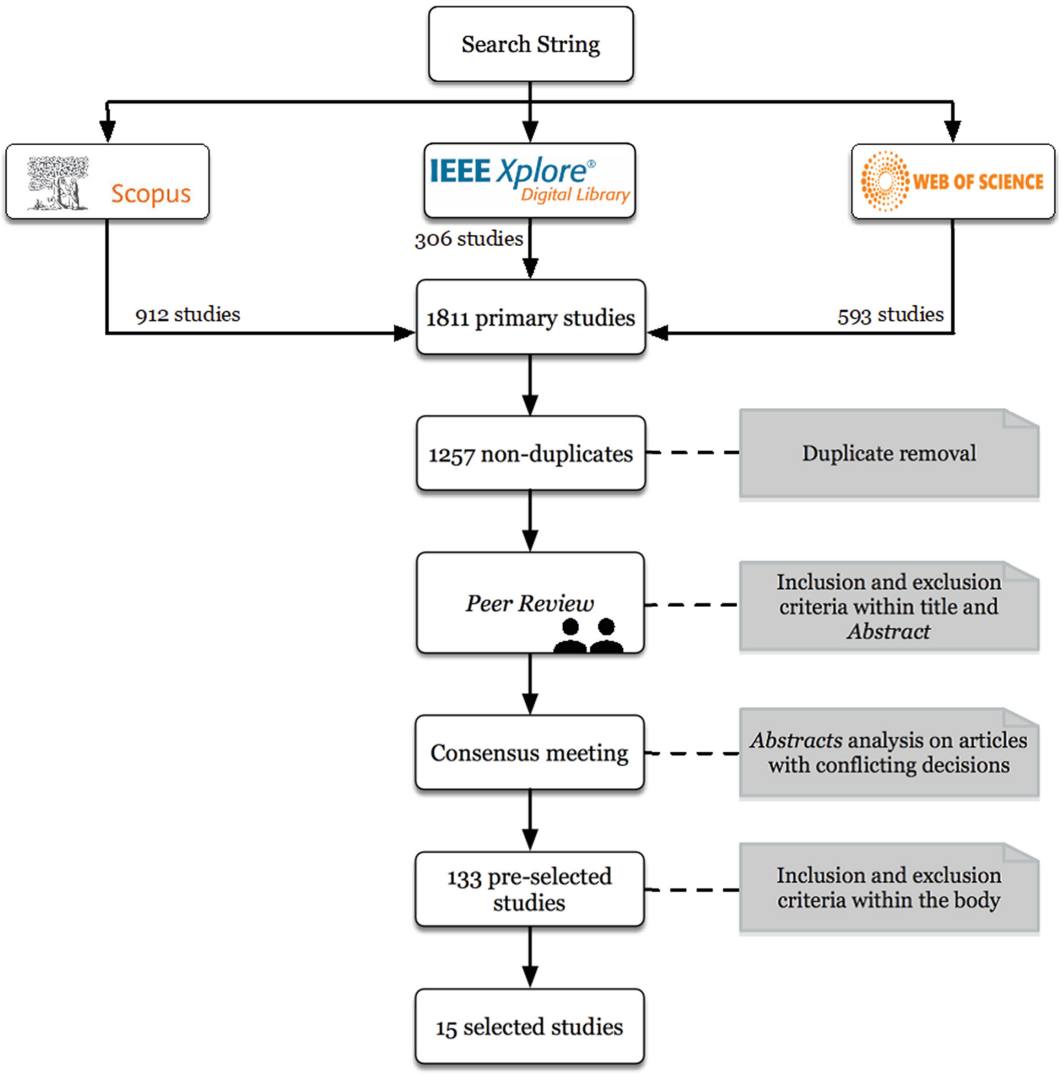


Fig. 1. Steps followed during the systematic mapping study.

4 Results and Discussion

Figure 2 synthesizes the results using two bubble scatter plots. The upper graph represents the number of articles published per year, according to publication type (journal, book chapter, or conference). Similarly, the lower graph plots the publication type against the classification tools (see Sect. 4.4). Thus, the bubbles are located at the intersections between the two axes and their size is proportional to the number of publications for each combination of values.

As can be seen in the upper part of Fig. 2, in 2016, only two studies were found. An excellent interest in tools that support the usability evaluation can be seen in 2017, where five of the 15 primary studies are concentrated. This interest progressively declines, finding three studies in 2018, two in 2019, and only one in 2020. Interest in this area of research recovers a little in 2021, with two studies.

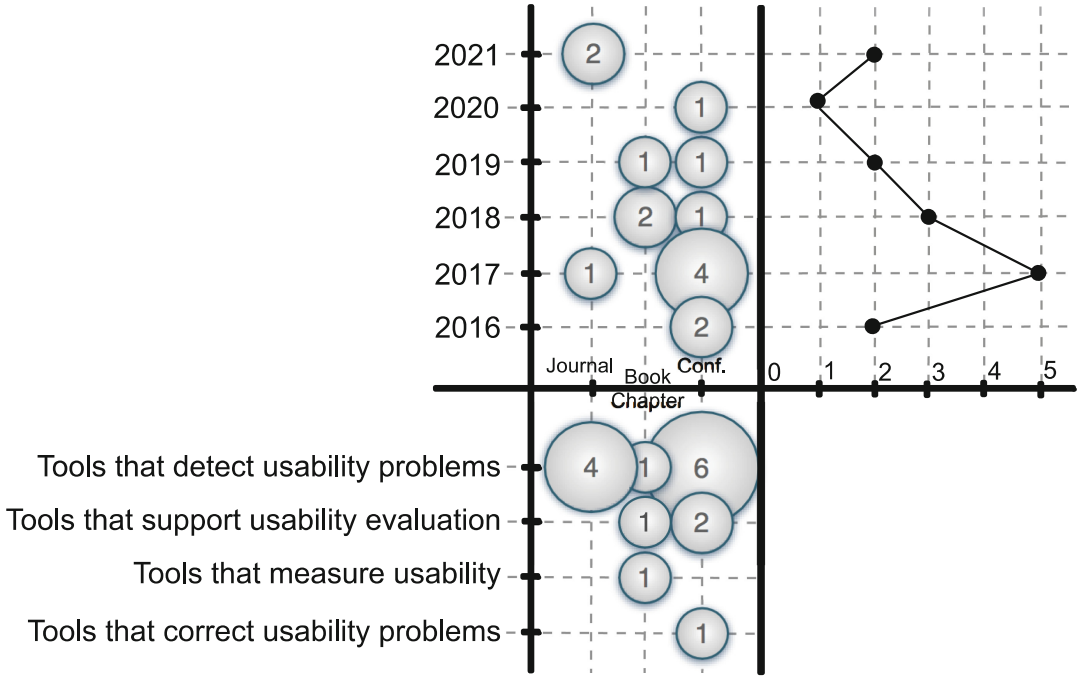


Fig. 2. Mapping for the primary study distribution between the classification of the tools along with the type of publication.

In the lower part of Fig. 2, it is seen that the classification that includes the most tools is "Tools that detect usability problems," followed by "Tools that support usability evaluation." Next, each research question will be answered.

4.1 Automated Tools to Support Usability Evaluation

In this section, we answer the first research question: *What are the automated tools that support the usability evaluation?* From the analysis of the 15 primary studies, 14 tools are obtained, which will be described below.

- **MOBILICS** [PS1] is an extension of USABILICS, so it inherits its methodology. This extension arises from the need to evaluate the usability of web pages in mobile environments considering the touch elements of these devices. MOBILICS performs the usability evaluation by comparing the actual interaction of a user performing a usability test with the interaction predefined by the evaluator who designs the test.
- **Environment for Supporting Interactive Systems Evaluation** [PS2] is a tool that automatically supports the usability evaluation of desktop web user interfaces. This tool performs usability evaluation by detecting usability problems through indicators, using usability data obtained from objective (e.g., an ergonomic guideline inspector) and subjective (through questionnaires) methods.
- **USF (Usability Smell Finder)** [PS3] is a tool that automatically supports usability evaluation of web user interfaces in desktop environments, operating as Software-as-a-Service (SaaS). This tool performs usability evaluation focusing on detecting usability smells, which serve as clues that point to possible usability problems.
- **MUSE (Mobile Usability Smell Evaluation)** [PS4] supports automatic usability evaluation of web user interfaces in desktop and mobile environments. MUSE records user interaction in usability testing sessions. Thanks to its proxy server approach, it can inject JavaScript code to the website to be evaluated without the need for the owner to do so manually.
- **Kobold** [PS5] supports automatic usability testing of web user interfaces in desktop environments, running as SaaS. This tool performs a usability evaluation focusing on detecting usability smells, providing refactorings that can be implemented manually, semi-automatically, or automatically to correct usability problems. Kobold is built on USF, so it uses a similar strategy when detecting usability smells.
- **Plain** [PS6] supports automatic usability evaluation of mobile applications. Plain is an Eclipse Plug-in that allows predicting the usability of a user interface by comparing usability metrics (e.g., composition, symmetry) with the properties of the elements that make up the mobile user interface to be evaluated.
- **UTAssistant** [PS7–PS9] allows supporting the usability evaluation automatically of user interfaces with a web focus. UTAssistant is a web platform that supports usability evaluation work by collecting mouse and keyboard log data during usability testing and allowing audio and video recording (both screen and user face).
- **Guideliner** [PS10] supports automatic usability evaluation of web user interfaces, both desktop, and mobile environments. Guideliner comprises several Java modules and uses Selenium WebDriver as its usability evaluation engine, allowing to search and analyze user interface elements and their features, comparing their values to guidelines to determine usability issues.
- **I2Evaluator** [PS11] supports automatic usability evaluation of web user interfaces, both mobile and desktop environments. i2evaluator seeks to measure user interfaces using aesthetic metrics (e.g., the balance of user interface objects) by incorporating an image decomposition algorithm that helps detect user interface elements to perform metric calculations.

- **PlatoS** [PS12] supports automatic usability evaluation of user interfaces in mobile application environments. The evaluator must create the tasks to perform in the usability tests and simulate the ideal interaction with the user interface. Using predefined usability metrics, PlatoS performs a statistical analysis of the times and actions performed by the evaluator and users to detect usability problems.
- **OwlEye** [PS13] supports automatic usability evaluation of mobile application user interfaces. OwlEye implements a CNN (Convolutional Neural Network) model for usability problem detection. With a set of 66,000 screenshots of more than 9,300 Android applications and using the CNN model, OwlEye can detect problems in user interfaces with a high level of efficiency.
- **ADUE (Automatic Domain Usability Evaluation)** [PS14] automatically allows usability evaluation of desktop applications. ADUE detects domain usability issues based on the domain usability approach. This approach covers aspects related more to the content of the elements that make up the user interface than to their characteristics. ADUE shows the tester the errors and associated components and provides recommendations to correct these problems.
- **GTmetrix** [PS15] supports automatic usability testing of web pages in desktop environments, detecting performance issues by comparing page metrics against 23 rules (related to performance aspects) provided by Yahoo. These values are compared with those detected on the page, and with this the problems to be solved are determined.
- **Dareboost** [PS15] supports the automatic usability evaluation of web pages in mobile environments, using performance metrics (e.g., load times) and comparing them with the metrics obtained from the elements of the web page to be analyzed. The tool provides reports showing the general score of the page, the number of problems, and the improvements recommended for their respective corrections.

4.2 Usability Evaluation Techniques Benefited by Automated Tools

This section answers the second research question: *Which usability evaluation-related techniques benefit from automated tools?* It is essential to mention that the tools have different functionalities and cover usability evaluation differently; therefore, the techniques benefited using these vary according to each case. The techniques used are described below.

- **Interaction Logging** is a technique that records the complete interaction of a user testing a system in such a way that it can be fully reproduced in real-time [25]. The tools that support this technique are Environment for Supporting Interactive Systems Evaluation [PS2], USF [PS3], MUSE [PS4], Kobold [PS5], UTAssistant [PS7–PS9], and PlatoS [PS12].
- **Standards Conformance Inspection** is an inspection method where technology specialists inspect the system determining whether it meets the previously proposed standards [25]. Tools that support this technique are Plain [PS6], I2Evaluator [PS11], PlatoS [PS12], and GTmetrix [PS15].
- **Questionnaires** are an indirect method for studying the user interface that allows knowing the user's opinions about the use of the interface but not giving direct information about it [1]. This technique is supported by two tools: Environment for Supporting Interactive Systems Evaluation [PS2] and UTAssistant [PS7–PS9].

- **Consistency Inspection** is a technique in which a team of designers inspects a set of interfaces for a family of products [25]. This technique is supported by two tools: Environment for Supporting Interactive Systems Evaluation [PS2] and ADUE [PS14].
- **Guidelines review** is a technique in which experts check the conformity of the user interface with the organizational guidelines document or with other guidelines [26]. This technique is supported by Guideliner [PS10] and GTmetrix [PS15].
- **Continuous Recording of User Performance** is a technique that emerges from the evaluation during the active use of the software that is intended to be evaluated [26]. The software architecture should make it easy for system administrators to collect data about system usage patterns, user performance speed, error rate, or frequency of online help replays. This technique is supported by the Environment for Supporting Interactive Systems Evaluation [PS2] and MUSE [PS4] tools.
- **Usage Logging** is a technique that seeks to record the user's actual usage in their interaction with a system, which implies having the computer automatically collect statistics about the detailed usage of the system [1]. This technique is supported by a single tool: MOBILICS [PS1].
- **Video/Audio Recording**, s its name suggests, seeks to generate audiovisual records of user interaction with systems in usability tests [27]. This technique is supported by a single tool: UTAssistant [PS7–PS9].
- **Time Keystroke Logging** is a technique that seeks to generate a record of each keystroke pressed by a user testing a system [25]. Each of these keystrokes is stored along with the exact time the event occurred. This technique is supported by a single tool: PlatoS [PS12].
- **Performance Metrics** is a technique in which essential aspects of the actual use of the software system to be evaluated are quantified, either in a controlled laboratory environment or in the usual work environment [28]. This technique is supported by a single tool: Dareboost [PS15].
- **Heuristic Evaluation** is a technique in which a usability expert observes an interface and tries to obtain an opinion on its good and bad characteristics [1]. This heuristic evaluation technique is supported by a single tool: OwlEye [PS13].

4.3 Problems and Challenges of Using Automated Tools

This section answers the third research question: *What are the existing problems and challenges of using automated tools for usability evaluation?* The main problems and challenges identified in the primary studies are described below.

- **Event Detection in Software Systems** is a technical problem based on the difficulties reported by the authors to identify when events occur in the systems to be evaluated. Goncalves et al. [PS1] reported that considering the MOBILICS tool, the main challenge was detecting events related to touch screens (i.e., *touchstar*, *touchmove* and *touchend*).
- **Detection of Indicators and Thresholds** can be considered an implementation difficulty when determining the metrics to use and how to capture the data that will make the corresponding comparisons with the user interface elements to be evaluated. Assilla et al. [PS2] referred to this problem in the context of the presentation of the Environment for Supporting Interactive Systems Evaluation tool.

- **Validation of Metrics** corresponds to the difficulty of choosing the metrics and quality standards so that the results of detecting usability problems are accurate. Generally, the tools that are based on these indicators must consider analysis models that allow detecting aspects of the user interface and translating them into values that can be interpreted and compared with these metrics. This is precisely the problem presented by the I2Evaluator tool [PS11].
- **A usability expert is still needed in some cases.** This is the main problem that tools seek to automate the usability evaluation. During their development, it must be determined how these tools will guarantee results that allow delivering a synthesis that defines the usability problems of a user interface. Avoiding depending on usability experts is one of the general problems [PS3].
- **General limitations and tool performance improvement.** It corresponds to the challenges reported in the primary studies [PS5, PS6, PS10]. Grigera et al. [PS5] considered improving the accuracy in detecting usability issues to automate new refactorings and select the most suitable one. Soui et al. [PS6] stated that some issues related to quality defect detection need to be investigated. In this way, the authors plan some refactoring operations (e.g., reorganization of the content of the mobile user interface). Marenkov et al. [PS10] specified the tool's limitations considering that it focuses on web environments, indicating that web pages that use Flash or Java Applets are not considered for the use of Guideliner.
- **Tools cannot completely replace manual evaluation.** The use of automated tools has several advantages, such as suitability for large-scale evaluation and less effort in terms of time. However, it is considered essential that these tools cannot completely replace manual tests since usability problems can be found but not how serious the problem is. Therefore, it is necessary to have a specific criterion that the evaluator must exercise based on the interpretation he wants to give to the information provided by the tool [PS15].

4.4 Classification of Automated Tools

In this section, the last research question is answered: *How can automated tools for usability evaluation be classified?* After analyzing the primary studies and the functionalities of each tool reported in each study, a total of four categories were identified, which will be described below.

- **Tools that measure usability.** The tools that belong to this category perform analysis of software systems and deliver an indicator (e.g., percentage of usability, a rating from 1 to 10) that describes the system's usability. These tools do not provide a very detailed analysis, nor do they provide feedback on the specific errors of the analyzed system in terms of usability. In this category, there is only the I2Evaluator tool [PS11].
- **Tools that support usability evaluation.** Tools belonging to this category perform analysis of software systems, provide an indicator that describes the usability of a system, and provide valuable functions that support the usability evaluation. Among these additional functions are (i) automated data capture (e.g., event log, log files), (ii) generation of forms for usability surveys, and (iii) timelines that support evaluation traceability usability, among others. In this category, there is only UTAssistant [PS7–PS9].

- **Tools that detect usability problems.** The tools that belong to this category perform an analysis of software systems and provide feedback on the specific errors found related to usability. The tool displays these errors, for example, in the form of alerts, reports, warnings. To this category belong the tools MOBILICS [PS1], Environment for Supporting Interactive Systems Evaluation [PS2], USF [PS3], MUSE [PS4], Plain [PS6], Guideliner [PS10], PlatoS [PS12], OwlEye [PS13], ADUE [PS14], GTmetrix [PS15] and Dareboost [PS15].
- **Tools that correct usability problems.** The tools that belong to this category analyze software systems and, in addition to providing feedback on the specific errors found in terms of usability, are given the option of correcting them automatically. Tools that perform error correction in a fully automated manner (without prior validation by the tool's user) or semi-automatically (the user authorizes the corresponding automatic correction) are considered in this category. In this category, there is only the Kobold tool [PS5].

5 Validity Threats

The first threat to validity is bias in the article selection process. The articles found with the search string used were evaluated according to the defined inclusion and exclusion criteria. Other researchers may have evaluated the publications differently. To corroborate the concordance in the selection of studies, meetings were held between the researchers to check the discarded preselected articles. Another aspect related to the selection of primary studies is the declared scope of our research since we only consider works published between 2016 and 2021. We may have missed some articles directly related to our research by only considering this period. We only consider the Scopus, IEEE Xplore, and WoS databases for the SMS performed. Although we found many results, more tools could have been reported in other databases. Another point regarding the scope of our research is that we do not consider the grey literature, which will most likely include results that are in line with the objective of this work.

6 Conclusions

A conclusion will be delivered according to each research question.

RQ1: What are the automated tools that support the usability evaluation?

According to the SMS carried out, it was possible to know the general panorama of the tools that support usability evaluation automatically reported in the literature. Between 2016 and 2021, 15 studies were found, of which 14 tools were identified. The reported tools show different approaches to support the usability evaluation. Note that these are presented with different methodologies and ways to support the evaluation of automated usability. The variety of reported tools spans desktop, mobile, and web applications (focused on desktop and mobile) that can be evaluated.

RQ2: Which usability evaluation-related techniques benefit from automated tools?

The most used technique is interaction recording, which makes sense since one of the most used approaches in tools is to perform usability tests to record the interaction of users with the evaluated interfaces. It can be noted that the tools focus on the methodology

they use according to the usability evaluation techniques. Some techniques reported [13, 14] were widely used (e.g., standards conformance inspection, consistency inspection), as well as techniques that were not addressed (e.g., collaborative usability inspection [28], pluralistic walkthrough [29]). This highlights that there is still work to be done to develop tools that support automatic usability evaluation. Covering the techniques that have not yet been addressed is a reason to encourage their development.

RQ3: What are the existing problems and challenges of using automated tools for usability evaluation?

According to what was identified in the primary studies, specific challenges, limitations, and problems can be highlighted when implementing the tools. One of these challenges is user interface event detection. This makes sense because it is the most important part of usability testing. Problems in detecting events in usability tests can cause erroneous results, which translates into poor usability for the evaluated interface. A similar aspect is that of the detection of indicators and thresholds. These must be defined and validated so that the tools, which focus on these aspects, can deliver a correct evaluation of usability. Although the tools greatly help the work of implementing an automated usability evaluation, usability experts are still needed, in some cases, to review the results [PS3, PS15].

RQ4: How can automated tools for usability evaluation be classified?

According to the analysis carried out on the primary studies identified in the SMS, the tools can be classified according to their approach and the functionalities that support automated usability evaluation. The classification of the tools is: (i) measure usability, (ii) support usability evaluation, (iii) detect usability problems, and (iv) correct usability problems.

The classification that includes the most tools correspond to those that detect usability problems. This is because it is a broader scope when facing a usability evaluation. These tools provide recommendations that guide the developers of the evaluated applications to correct the usability errors detected. A broader scope is that the tool automatically detects usability problems; only one tool belongs to this category. Kobold [PS5] is presented as one of the most exciting tools because it integrates automatic and semi-automatic refactoring of web user interface elements.

As future works, we will consider more databases (e.g., ACM Digital Library, SpringerLink). Additionally, study and consider the grey literature to expand the results when looking for tools that support the usability evaluation automatically. We want to explore tools that perform qualitative usability evaluations [30]. Finally, we expect to study the usability evaluations results to prioritize and recommend the most relevant aspects [31].

Acknowledgment. This work was supported by the Chilean Ministry of Education and the University of Atacama (ATA1899 project).

Appendix A: Primary Studies

This appendix lists the references of the primary studies used for the mapping study described in this paper.

[PS1] Gonçalves, L. F., Vasconcelos, L. G., Munson, E. V., Baldochi, L. A.: Supporting adaptation of web applications to the mobile environment with automated usability evaluation. In: 31st Annual ACM Symposium on Applied Computing (SAC'16), ACM, Pisa, Italy, pp. 787–794 (2016). https://doi.org/10.1145/2851613.2851863.

[PS2] Assila, A., de Oliveira, K. M., Ezzedine, H.: An environment for integrating subjective and objective usability findings based on measures. In: 2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS'16), IEEE, Grenoble, France, pp. 1–12 (2016). https://doi.org/10.1109/RCIS.2016.7549320.

[PS3] Grigera, J., Garrido, A., Rivero, J. M., Rossi, G.: Automatic detection of usability smells in web applications. International Journal of Human-Computer Studies 97, 129–148 (2017a). https://doi.org/10.1016/j.ijhcs.2016.09.009.

[PS4] Paternò, F., Schiavone, A. G., Conti, A.: Customizable automatic detection of bad usability smells in mobile accessed web applications. In: 19th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI'17), ACM, Vienna, Austria, article 42, pp. 1–11 (2017). https://doi.org/10.1145/3098279.3098558.

[PS5] Grigera, J., Garrido, A., Rossi, G.: Kobold: web usability as a service. In: 2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE'17), Urbana, IL, USA, pp. 990–995 (2017). https://doi.org/10.1109/ASE.2017.8115717.

[PS6] Soui, M., Chouchane, M., Gasmi, I., Mkaouer, M. W.: PLAIN: PLugin for predicting the usability of mobile user interface. In: 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP'17) - Vol. 1: GRAPP, Porto, Portugal, pp. 127–136 (2017). https://doi.org/10.5220/0006171201270136.

[PS7] Desolda, G., Gaudino, G., Lanzilotti, R., Federici, S., Cocco, A.: UTAssistant: A web platform supporting usability testing in italian public administrations. In: 12th Biannual Conference of the Italian SIGCHI Chapter (CHItaly'17), Cagliari, Italy, pp. 138–142 (2017).

[PS8] Federici, S., Mele, M. L., Lanzilotti, R., Desolda, G., Bracalenti, M., Meloni, F., Gaudino, G., Cocco, A., Amendola, M.: UX evaluation design of UTAssistant: A new usability testing support tool for italian public administrations. In: Kurosu M. (ed.) Human-Computer Interaction. Theories, Methods, and Human Issues. HCI 2018 (55–67). Lecture Notes in Computer Science, vol 10901. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91238-7_5.

[PS9] Federici, S., Mele, M. L., Bracalenti, M., Buttafuoco, A., Lanzilotti, R., Desolda, G.: Bio-behavioral and self-report user experience evaluation of a usability assessment platform (UTAssistant). In: 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP'19) - Vol. 2: HUCAPP, Prague, CZ, pp. 19–27 (2019).

[PS10] Marenkov, J., Robal, T., Kalja, A.: Guideliner: A tool to improve web UI development for better usability. In: 8th International Confe- rence on Web Intelligence, Mining and Semantics (WIMS'18), ACM, Novi Sad, Serbia, article 17, pp. 1–9 (2018). https://doi.org/10.1145/3227609.3227667.

[PS11] Chettaoui, N. Bouhlel, M. S.: I2Evaluator: An aesthetic metric-tool for evaluating the usability of adaptive user interfaces. In: Hassanien, A. E., Shaalan, K., Gaber, T., and Tolba, M. F. (eds.) Proceedings of the International Conference on Advanced Intelligent Systems and Informatics. AISI 2017 (374–383). Advances in Intelligent Systems and Computing, vol 639. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64861-3_35.

[PS12] Barra, S., Francese, R., Risi, M.: Automating mockup-based usability testing on the mobile device. In: Miani, R., Camargos, L., Zarpelão, B., Rosas, E., and Pasquini, R. (eds.) Green, Pervasive, and Cloud Computing. GPC 2019 (128–143). Lecture Notes in Computer Science, vol 11484. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-19223-5_10.

[PS13] Liu, Z., Chen, C., Wang, J., Huang, Y., Hu, J., Wang, Q.: Owl eyes: Spotting UI display issues via visual understanding. In: 35th IEEE/ACM International Conference on Automated Software Engineering (ASE'20), ACM, Virtual Event Australia, pp. 398–409 (2020). https://doi.org/10.1145/3324884.3416547.

[PS14] Bacíková, M., Porubän, J., Sulír, M., Chodarev, S., Steingartner, W., Madeja, M.: Domain usability evaluation. Electronics 10(16), 1–28, article 1963, (2021). Doi: 10.3390/ electronics10161963.

[PS15] Al-Sakran, H. O. Alsudairi, M. A.: Usability and accessibility assessment of saudi arabia mobile E-government websites. IEEE Access 9, 48254–48275 (2021). https://doi.org/10.1109/ACCESS.2021.3068917.

References

1. Nielsen, J.: Usability engineering. Morgan Kaufmann Publishers Inc., San Francisco (1994).ISBN: 978-0080520292
2. ISO 9241–11:2018. Ergonomics of human-system interaction–part 11: Usability: Definitions and concepts, ISO (2018)
3. Ferré, X.: Marco de integración de la usabilidad en el proceso de desarrollo software. Facultad de Informática, Universidad Politécnica de Madrid, Madrid, Spain, Tesis doctoral (2005)
4. Ivory, M.Y., Hearst, M.A.: The state of the art in automating usability evaluation of user interfaces. ACM Comput. Surv. **33**(4), 470–516 (2001). https://doi.org/10.1145/503112.503114
5. Marenkov, J., Robal, T., Kalja, A.: Guideliner: a tool to improve web UI development for better usability. In: 8th International Conference on Web Intelligence, Mining and Semantics (WIMS 2018), ACM, Novi Sad, Serbia, article 17, pp. 1–9 (2018). https://doi.org/10.1145/3227609.3227667
6. Fabo, P., Durikovic, R.: Automated usability measurement of arbitrary desktop application with eyetracking. In: 2012 16th International Conference on Information Visualisation, IEEE, Montpellier, France, pp. 625–629 (2012). https://doi.org/10.1109/IV.2012.105
7. Federici, S., et al.: UX evaluation design of UTAssistant: a new usability testing support tool for Italian public administrations. In: Kurosu, M. (ed.) HCI 2018. LNCS, vol. 10901, pp. 55–67. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91238-7_5
8. Grigera, J., Garrido, A., Rossi, G.: Kobold: web usability as a service. In: 2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE 2017), IEEE, Urbana, IL, USA, pp. 990–995 (2017). https://doi.org/10.1109/ASE.2017.8115717

9. Liyanage, N. L., Vidanage, K.: Site-ability: a website usability measurement tool. In: 2016 Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer'16), IEEE, Negombo, Sri Lanka, pp. 257–265 (2016). Doi: https://doi.org/10.1109/ICTER.2016.7829929
10. Charfi, S., Trabelsi, A., Ezzedine, H., Kolski, C.: Widgets dedicated to user interface evaluation. Int. J. Hum.-Comput. Interact. **30**(5), 408–421 (2014). https://doi.org/10.1080/10447318.2013.873280
11. Bakaev, M., Mamysheva, T., Gaedke, M.: Current trends in automating usability evaluation of websites: can you manage what you can't measure? In: 2016 11th International Forum on Strategic Technology (IFOST 2016), Novosibirsk, Russia, pp. 510–514. IEEE (2016). https://doi.org/10.1109/IFOST.2016.7884307
12. Khasnis, S. S., Raghuram, S., Aditi, A., Samrakshini, R. S., Namratha, M.: Analysis of automation in the field of usability evaluation. In: 2019 1st International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE 2019), Bangalore, India, pp. 85–91. IEEE (2019). https://doi.org/10.1109/ICATIECE45860.2019.9063859
13. Ferré, X., Juristo, N., Moreno, A.M.: Deliverable D.5.1. selection of the software process and the usability techniques for consideration. STATUS Project (code IST-2001–32298) financed by the European Commission from December of 2001 to December of 2004 (2002). http://is.ls.fi.upm.es/status/results/STATUSD5.1v1.0.pdf
14. Ferré, X., Juristo, N., Moreno, A. M.: Deliverable D.5.2. specification of the software process with integrated usability techniques. STATUS Project (code IST-2001–32298) financed by the European Commission from December of 2001 to December of 2004 (2002). http://is.ls.fi.upm.es/status/results/STATUSD5.2v1.0.pdf
15. Kitchenham, B.A., Budgen, D., Brereton, O.P.: Using mapping studies as the basis for further research–a participant-observer case study. Inf. Softw. Technol. **53**(6), 638–651 (2011). https://doi.org/10.1016/j.infsof.2010.12.011
16. Zhang, H., Babar, M.A., Tell, P.: Identifying relevant studies in software engineering. Inf. Softw. Technol. **53**(6), 625–637 (2011). https://doi.org/10.1016/j.infsof.2010.12.010
17. Assila, A., de Oliveira, K. M., Ezzedine, H.: An environment for integrating subjective and objective usability findings based on measures. In: 2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS 2016), Grenoble, France, pp. 1–12. IEEE (2016). https://doi.org/10.1109/RCIS.2016.7549320
18. Barra, S., Francese, R., Risi, M.: Automating Mockup-based usability testing on the mobile device. In: Miani, R., Camargos, L., Zarpelão, B., Rosas, E., Pasquini, R. (eds.) GPC 2019. LNCS, vol. 11484, pp. 128–143. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-19223-5_10
19. Paternò, F., Schiavone, A. G., Conti, A.: Customizable automatic detection of bad usability smells in mobile accessed web applications. In: 19th International Conference on Human-Computer Interaction with Mobile Devices and Services (Mo-bileHCI 2017), Vienna, Austria, article 42, pp. 1–11. ACM (2017). https://doi.org/10.1145/3098279.3098558
20. Atlas.ti9 Atlas.ti 9 desktop trial (windows) (2021). https://atlasti.com/
21. Scopus.com: An eye on global research: 5000 Publishers. Over 71 M records and 23,700 titles 2020. https://www.scopus.com/freelookup/form/author.uri. Accessed 16 Sept 21
22. Castro, J. W., Acuña, S. T.: Comparativa de selección de estudios primarios en una revisión sistemática. In: XVI Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2011), A Coruña, España, pp. 319–332 (2011). http://hdl.handle.net/10486/665299. Accessed 16 Sept 21
23. Magües, D., Castro, J.W., Acuña, S.T.: Usability in agile development: a systematic mapping study. In: XLII Conferencia Latinoamericana de Informática (CLEI 2016), Valparaíso, Chile, pp. 677–684. IEEE (2016). https://doi.org/10.1109/CLEI.2016.7833347

24. Ren, R., Castro, J.W., Acuña, S.T., De Lara, J.: Evaluation techniques for chatbot usability: a systematic mapping study. Int. J. Software Eng. Knowl. Eng. **29**(11n12), 1673–1702 (2019). https://doi.org/10.1142/S0218194019400163
25. Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., Carey, T.: Human-Computer Interaction. Concepts and Design. Addison-Wesley, Harlow (1994). ISBN: 978–0201627695
26. Shneiderman, B.: Designing the User Interface: Strategies for Effective Human-Computer. Pearson, Boston (1998). ISBN: 978–0201694970
27. Hix, D., Hartson, H.R.: Developing User Interfaces: Ensuring Usability Through Product & Process. Wiley, New York (1993).ISBN: 978-0471578130
28. Constantine, L.L., Lockwood, L.A.: Software for use: A Practical Guide to the Models and Methods of Usage-Centered Design. Addison-Wesley Professional, New York (1999).ISBN: 978-0321773722
29. Nielsen, J.: Usability inspection methods. In: Conference Companion on Human Factors in Computing Systems (CHI 1994), Boston, Massachusetts, USA, pp. 413–414. ACM (1994). https://doi.org/10.1145/259963.260531
30. Rojas P., L.A., Truyol, M.E., Calderon Maureira, J.F., Orellana Quiñones, M., Puente, A.: Qualitative evaluation of the usability of a web-based survey tool to assess reading comprehension and metacognitive strategies of university students. In: Meiselwitz, G. (ed.) HCII 2020. LNCS, vol. 12194, pp. 110–129. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49570-1_9
31. Rojas, L.A., Macías, J.A.: Toward collisions produced in requirements rankings: a qualitative approach and experimental study. J. Syst. Softw. **158**, 110417 (2019). Article 42. https://doi.org/10.1016/j.jss.2019.110417

Apéndice E - Herramientas Automatizadas para la Evaluación de la Usabilidad

En capítulos anteriores se presentaron los resultados de la investigación al realizar un SMS con el objetivo de identificar herramientas que permitan la evaluación automática de la usabilidad. Considerando las herramientas reportadas en la Sección 5.1, se describirá en el presente apéndice cada una de estas herramientas. Se explicará su funcionamiento a grandes rasgos, la forma en la que apoyan la usabilidad y los detalles más relevantes. Cabe destacar que estas herramientas están ordenadas según la fecha de publicación de los estudios primarios correspondientes.

E.1. MOBILICS

MOBILICS es una extensión de una herramienta existente llamada USABILICS [Gonçalves *et al.*, 2016]. Esta herramienta pertenece a la categoría de Herramientas que detectan problemas de usabilidad.

MOBILICS surge de la necesidad de ayudar a los desarrolladores a adaptar sitios web de escritorio al entorno móvil. Al ser una extensión de USABILICS, realiza una evaluación automatizada de la usabilidad basada en tareas de aplicaciones web de escritorio convencionales, siendo estas llevadas a cabo en el entorno móvil gracias a MOBILICS.

De forma simplificada, Gonçalves *et al.* [2016] explican que la herramienta funciona definiendo tareas para una aplicación web determinada que se quiera evaluar. Luego, se capturan las interacciones de los usuarios finales que realizan tests de usabilidad siguiendo las tareas definidas previamente utilizando los dispositivos mó-

viles. Las interacciones del usuario se comparan con las tareas definidas. MOBILICS detecta problemas de usabilidad señalando las tareas que presentaron problemas (interpretados como problemas de usabilidad) y los elementos de la interfaz donde se detectaron estos problemas. Además de detectar estos problemas, la herramienta puede entregar recomendaciones con el objetivo de mejorar la usabilidad móvil de los elementos que conforman la interfaz.

Como la herramienta mencionada está basada en USABILICS, hereda sus actividades: (i) Definición de tareas, (ii) *Logging*, (iii) Análisis de tareas y (iv) Recomendaciones. En la definición de tareas, USABILICS implementa una sub-herramienta llamada UseTasker, que permite a los desarrolladores definir tareas simplemente interactuando con la IU de la aplicación. Para definir una tarea, basta con utilizar la aplicación como se espera del usuario final. En la actividad de *Logging*, USABILICS incrusta en las páginas web una aplicación cliente en JavaScript que reconoce todos los elementos de la página web usando el Modelo de Objetos de Documento (DOM o *Document Object Model* por sus siglas en inglés) vinculando eventos a estos elementos, permitiendo la recopilación de interacciones del usuario como movimientos del mouse, desplazamiento, cambio de tamaño de ventana, carga y descarga de la página, entre otros. La aplicación cliente comprime los registros y los envía a una aplicación servidor que almacena los datos en una base de datos relacional para ser utilizados durante la fase de análisis de tareas. El análisis de tareas se realiza comparando la secuencia de eventos registrados por una tarea determinada y la secuencia de eventos correspondiente a la interacción del usuario. La similitud entre estas interacciones proporciona una métrica de eficiencia. El porcentaje de finalización de una tarea proporciona una métrica de efectividad. Luego de realizar estas comparaciones y obtener estas métricas, la herramienta puede determinar tres situaciones diferentes que indican la ocurrencia de acciones incorrectas: (i) una acción no pertenece a la secuencia correcta de acciones, (ii) una acción se omite de la secuencia, (iii) una acción dentro de la secuencia es reemplazada por otra acción. En la fase de recomendaciones, la herramienta puede analizar estas tres tareas diferentes que presentan baja usabilidad para así poder entregar una recomendación.

MOBILICS hereda la metodología explicada anteriormente. Esta extensión de la herramienta USABILICS incluye otras consideraciones, ya que el objetivo de esta es poder evaluar la usabilidad de entornos móviles. El principal aspecto considerando para poder generar la compatibilidad con este enfoque es considerar los Eventos

Touch. Al ser la principal diferencia entre el entorno de escritorio y el entorno de móvil, se debe contar con estos métodos de entrada de datos. Los eventos considerados son *touchstart*, *touchmove* y *touchend* (correspondientes al inicio, el movimiento y el final de una interacción *touch* respectivamente). Eventos como los pellizcos se desprenden de estos eventos, ya que se detectan como la ocurrencia de dos eventos de *touchstart* y *touchmove* simultáneos. Estos eventos son implementados de la misma forma por la herramienta, siendo una aplicación cliente en JavaScript la que registra los eventos para enviarlos a una aplicación servidor para el análisis de los datos.

Para la **fase de definición de las tareas**, la aplicación UsaTasker fue mejorada para poder crear los eventos de la misma forma en la que se crean con USABILICS, pero contando con los eventos asociados al entorno web móvil. En la Figura E.1 se puede apreciar el uso de UsaTasker a la hora de definir estas tareas. Una vez que la tarea se ha definido, se desplegará un menú de opciones para generalizar la tarea y objetos de la página con una opción de opacidad que permite al usuario ver el contenido de la página web donde se realiza la tarea e interacción. En el artículo se explica con más detalle esta función [Gonçalves *et al.*, 2016].

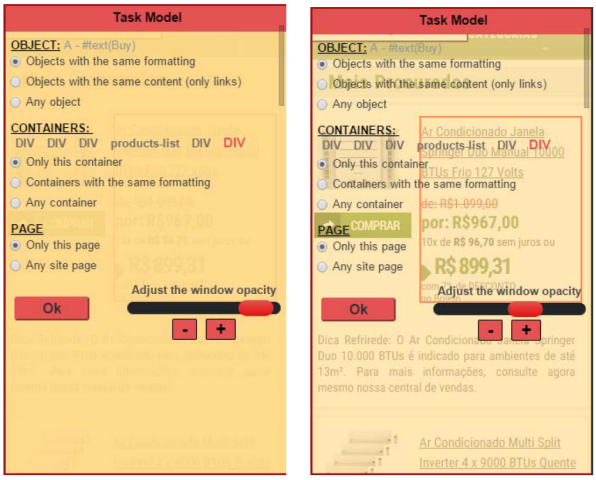


Figura E.1: UsaTasker mostrando las opciones de la tarea definida. Opacidad normal (izquierda) y baja opacidad (derecha) [Gonçalves *et al.*, 2016].

Cuando el evaluador finaliza la definición de tareas, UsaTasker presenta los eventos capturados gráficamente como se muestra en la Figura E.2. Esta visualización proporciona una forma de verificar si cada evento que compone una tarea fue registrado correctamente. En la Figura E.2 cada cuadro representa un evento y las

flechas rojas indican el orden de cada evento dentro de la misma tarea. En esta misma visualización el evaluador puede gestionar las tareas.

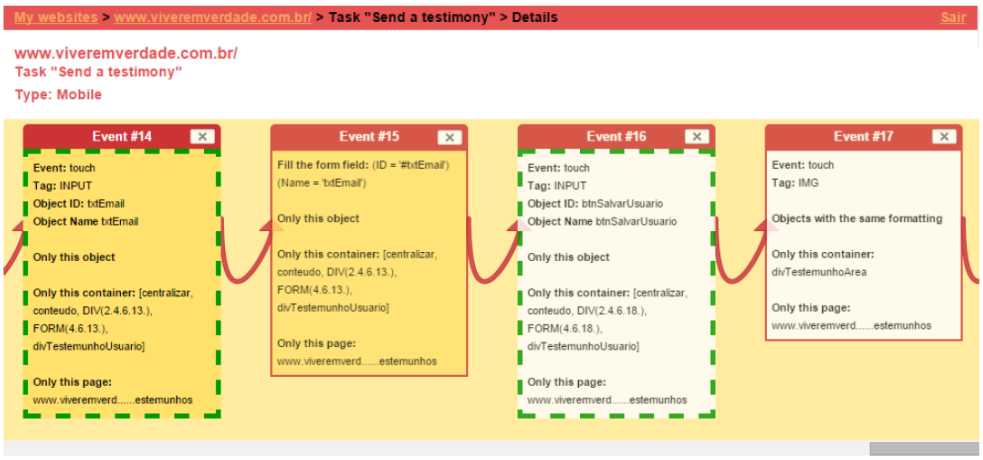


Figura E.2: Eventos capturados con MOBILICS [Gonçalves *et al.*, 2016].

La **fase de *Logging*** es similar a la de USABILICS. La principal diferencia está relacionada con el hecho de que los móviles pueden activar eventos *touch*. Es por esto que solo se filtran estos eventos (no se consideran los eventos de mouse debido al enfoque móvil) reduciendo así el volumen de datos que deben registrarse. Estos datos se comprimen y se envían al servidor, al igual que con USABILICS.

En la **fase de análisis de tareas** se realizan los cambios en función de los eventos *touch* explicados anteriormente con el fin de realizar las comparaciones de las métricas con base en estos componentes. Como se mencionó para USABILICS, la forma de detectar los problemas de usabilidad es mediante la comparación de la interacción del usuario real con la tarea definida por el evaluador. Esta comparación produce un índice de comparación entre 0 y 1. Entre más cercano es este valor al 1, más similitud existe entre la interacción del usuario y la tarea definida.

En la **fase de recomendaciones**, MOBILICS usa los índices obtenidos con base en las tareas realizadas y los elementos que conforman la actividad diseñada. Luego de presentar estos índices y sus correspondientes tareas, MOBILICS genera un informe que contiene las recomendaciones cuando se detectan problemas de usabilidad y se necesiten corregirlas. El informe también presenta enlaces a contenido disponible en la web que abordan buenas prácticas de programación que mejoran la usabilidad de las aplicaciones web y móviles.

E.2. Environment for Supporting Interactive Systems Evaluation

Esta herramienta se presenta como un entorno el cuál integra cuatro sub-herramientas, de las cuales tres se encargan de evaluar la interacción del usuario y evaluar la usabilidad ergonómica de la IU, además de generar y gestionar cuestionarios [Assila *et al.*, 2016]. Una cuarta herramienta se encarga de sintetizar los resultados de las herramientas anteriores. Esta herramienta tiene un enfoque centrado en aplicaciones web de escritorio, y pertenece a la categoría de Herramientas que detectan problemas de usabilidad.

Este entorno surge de la necesidad de integrar los métodos para implementar una evaluación de la usabilidad correspondiente a enfoques subjetivos y objetivos, según señalan Assila *et al.* [2016]. Este entorno proporciona una forma automatizada de recopilar y analizar datos de usabilidad a través de la unificación de tres herramientas que respaldan una evaluación objetiva y subjetiva de interfaces de usuario. Para asegurar la integración, los autores señalan que aplican los conceptos de medidas propuestos por la norma ISO/IEC 15939 [ISO/IEC, 2007].

Como se mencionó anteriormente, este entorno se compone de cuatro sub-herramientas que, integradas de acuerdo a lo señalado por Assila *et al.* [2016], permiten apoyar la evaluación de la usabilidad de forma automática. Estas herramientas son las siguientes:

- Un informador electrónico llamado *Environment for Interactive System Evaluation* (*IESEval*) que se ocupa de la evaluación de la interacción entre los usuarios y el sistema evaluado.
- Una herramienta generadora de cuestionarios que soporta una evaluación subjetiva para evaluar las percepciones de los usuarios sobre el sistema evaluado.
- Un inspector de pautas ergonómicas que garantiza una evaluación de la usabilidad ergonómica de las interfaces de usuario.
- Una herramienta de síntesis de los resultados de la evaluación que permite integrar los datos de usabilidad de las tres herramientas previamente mencionadas para proporcionar una buena medida de usabilidad.

Este entorno contribuye a evaluar la usabilidad de las interfaces de usuario de los sistemas interactivos con base en los aspectos de las tres primeras herramientas mencionadas (*IESEval*, la herramienta generadora de cuestionarios y el inspector de pautas ergonómicas) sobre evaluación subjetiva y objetiva, automatizar recogida de datos de usabilidad de las herramientas y también el análisis de los resultados, integrar estos resultados de usabilidad, detectar con precisión problemas de usabilidad, apoyar la interpretación de los resultados para la toma de decisiones en materia de corrección de problemas de usabilidad y generar recomendaciones para mejorar interfaces de usuario.

En la Figura E.3 se aprecia una vista general de cómo se conforma este entorno. Se pueden apreciar las tres herramientas primarias (*IESEval*, la herramienta generadora de cuestionarios y la herramienta de inspección de usabilidad ergonómica) en paralelo. La herramienta generadora de cuestionarios puede integrarse con *IESEval* y con la herramienta de inspección de usabilidad ergonómica. Los datos generados por todas estas herramientas son tomados por la herramienta de síntesis de evaluación de resultados, entregando los datos resultantes al evaluador.

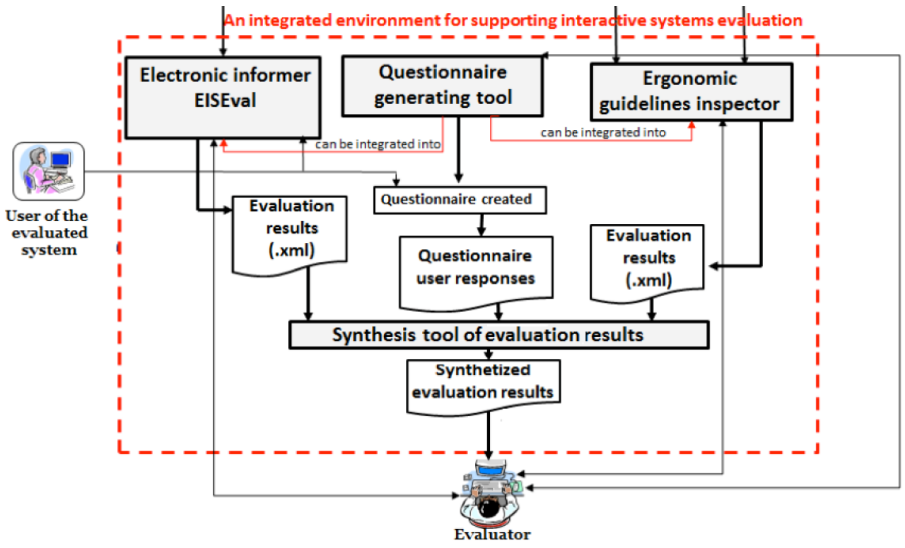


Figura E.3: Una vista general de la construcción de la herramienta presentada como Environment for Supporting Interactive Systems Evaluation [Assila *et al.*, 2016].

IESEval [Assila *et al.*, 2016] es un sistema interactivo que se basa en la evaluación objetiva y se enfoca en evaluar la interacción del usuario con el sistema evaluado. Permite capturar y analizar las acciones de los usuarios y sus interacciones con el

sistema interactivo evaluado en tiempo real. Esta herramienta permite capturar un conjunto de medidas base útiles para algunos criterios de usabilidad como efectividad, eficiencia y *minimal actions*. Algunas de estas medidas son el número de tareas realizadas, tiempo para realizar una tarea, número de tareas realizadas con éxito, número de tareas fallidas, número de acciones realizadas por un usuario, entre otras.

Según lo explicado por Assila *et al.* [2016], se desarrolla una herramienta de generación de cuestionarios de usabilidad, estandarizados o personalizados, que permiten asegurar una evaluación subjetiva de los sistemas interactivos. Esta herramienta permite capturar las percepciones de los usuarios sobre la usabilidad del sistema en general o sobre criterios de usabilidad más específicos. Existe una fase de análisis automático para apoyar a los evaluadores en la detección de problemas de la IU según diferentes criterios de usabilidad (como la eficacia, eficiencia, legibilidad, etc.).

Según Assila *et al.* [2016], se desarrolla un inspector de pautas ergonómicas para evaluar la consistencia de la IU a evaluar. Este inspector se basa en un conjunto de pautas ergonómicas que permiten detectar inconsistencias en las interfaces de usuario y brindan una lista de recomendaciones. Se pueden capturar varias medidas base relacionadas con los criterios de usabilidad ergonómica. Entre estas medidas detectadas se encuentran el número de tamaños y colores de fuente, color de fondo, dimensiones de los componentes, dimensiones de imagen, dimensiones de pantalla, número de funciones de la interfaz, entre otros aspectos.

La cuarta herramienta, que corresponde a la herramienta de síntesis de los resultados de la evaluación, toma los resultados de las tres herramientas explicadas anteriormente y los sintetiza para entregar resultados de usabilidad. Esta se enfoca en aplicaciones web de escritorio y consiste de tres módulos principales: (i) Módulo de preparación, (ii) Módulo de evaluación y (iii) Módulo de análisis de resultados. La arquitectura de la herramienta de síntesis de resultados de evaluación se presenta en la Figura E.4.

El **módulo de preparación** permite preparar la evaluación, incluyendo todos los componentes necesarios. Estos componentes son la gestión de criterios (que trata los criterios de calidad que se abordarán en la evaluación de la usabilidad), la gestión de medidas (que contiene todas las medidas predefinidas ya adoptadas en este entorno para construir los indicadores de usabilidad) y la gestión de cuestionarios (que incluye todas las funcionalidades de la herramienta de generación de cuestionarios). Todos estos datos se guardan en una base de datos.

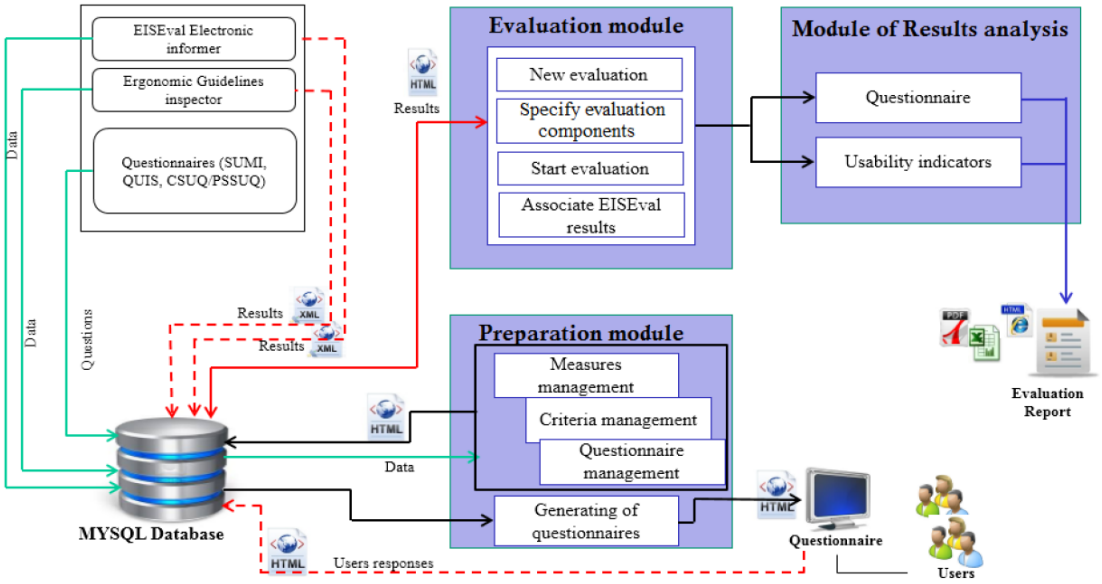


Figura E.4: Arquitectura de la herramienta de síntesis de los resultados de la evaluación [Assila *et al.*, 2016].

El **módulo de evaluación** permite evaluar la usabilidad de un sistema dado de acuerdo con un conjunto de criterios seleccionados para detectar si hay problemas en las interfaces de usuario. Este módulo consta de cuatro componentes: (i) la creación de una nueva evaluación (en la que se adhieren todos los elementos necesarios para iniciar la evaluación como los criterios de usabilidad a evaluar, el código fuente del sistema, etc.), (ii) la especificación de los componentes de la evaluación (que consiste en la creación de la lista de tareas a realizar por los usuarios durante los tests de usabilidad, las recomendaciones ergonómicas seleccionadas sobre algunas medidas de usabilidad y el cuestionario seleccionado para ser utilizado en la evaluación), (iii) el inicio de la evaluación (donde el evaluador procede a la sesión de evaluación involucrando a los usuarios para la captura de todos los datos relacionados con las interacciones de sistema y percepciones del mismo) y (iv) asociar los resultados dados por *IESEval* (que trata de la asociación de todos los resultados dados de *IESEval* con sus correspondientes usuarios).

El **módulo de análisis de resultados** proporciona el análisis y la síntesis de todos los resultados de la evaluación y luego proporciona diferentes formatos para generarlos. Consta de dos componentes: (i) los resultados del cuestionario (generados en excel, HTML o PDF que retratan las percepciones del usuario de acuerdo con los cuestionarios completados, generando análisis de resultados, criterios de evaluación

e información de problemas de usabilidad detectados) y (ii) los indicadores de usabilidad (que realiza la integración de los datos de usabilidad (subjetivos y objetivos) de manera complementaria y consistente).

Los indicadores de usabilidad, de acuerdo con lo explicado anteriormente, forman parte fundamental del módulo de análisis de resultados de la herramienta de síntesis. Estos indicadores se pueden apreciar en la Figura E.5, que muestra la interfaz de la herramienta final de síntesis de los resultados de la evaluación. En el menú lateral se pueden apreciar las funciones que permiten definir y comprobar los criterios, indicadores y medidas para la detección de problemas de usabilidad. Se aprecia también una selección de indicadores que se quiere comprobar. En este caso, se selecciona la densidad de información, indicando en el gráfico los puntos que corresponden a elementos con problemas de usabilidad. Estos elementos se nombran de acuerdo con los componentes de la IU que tienen dicho problema, brindando recomendaciones en el proceso.

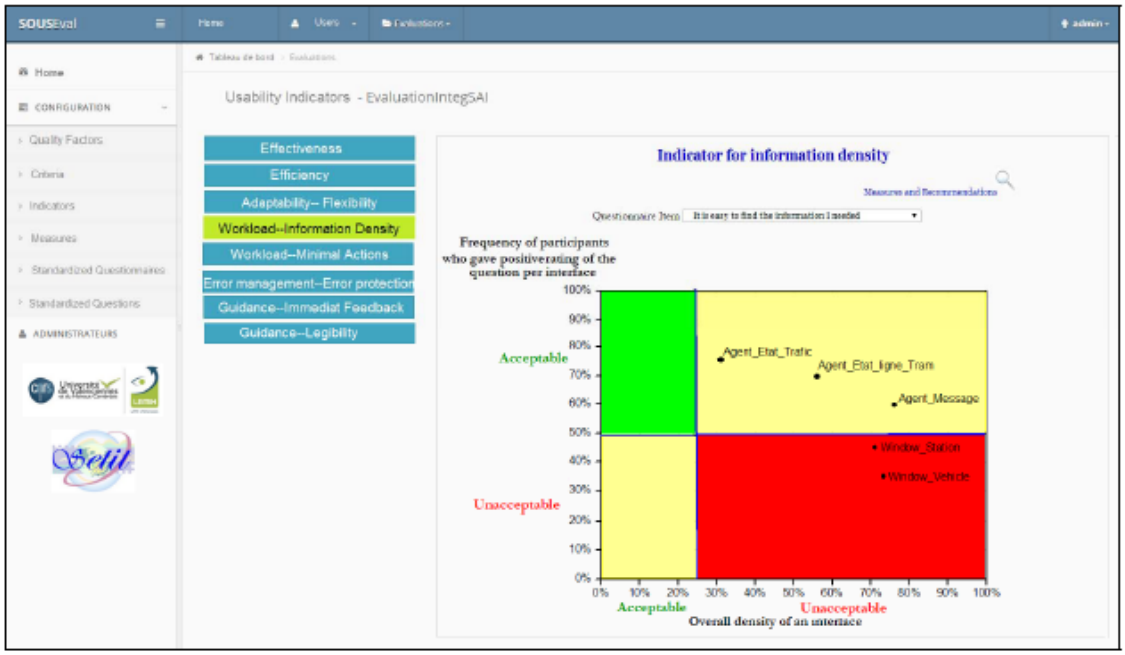


Figura E.5: Presentación final de los indicadores de información para la herramienta de síntesis de los resultados de la evaluación [Assila *et al.*, 2016].

E.3. USF (*Usability Smell Finder)*

USF es una herramienta que apoya la evaluación de la usabilidad detectando problemas de usabilidad en páginas web enfocadas a entornos de escritorio [Grigera *et al.*, 2017a]. Esta herramienta pertenece a la clasificación de Herramientas que detectan problemas de usabilidad.

USF puede ser utilizado como SaaS (Software como Servicio o *Software-as-a-Service* por sus siglas en inglés) permitiendo, con un esfuerzo de configuración mínimo, proporcionar consejos y recomendaciones de corrección para los problemas de usabilidad detectados en aplicaciones web. Por lo tanto, está dirigido a profesionales con diferentes niveles de experiencia en materias de usabilidad. Por un lado, los expertos en usabilidad pueden utilizar USF para obtener una retroalimentación rápida de las interacciones reales de una masa de usuarios, configurando la herramienta según sus necesidades. Por otro lado, los desarrolladores sin experiencia en materias de usabilidad pueden usar USF luego de una instalación simple, dejar que la herramienta recopile evidencia para detectar y diagnosticar problemas de usabilidad e implementar las soluciones que esta sugiera.

La metodología utilizada por esta herramienta se centra en la detección de *Usability Smells* lo cual, descrito por Grigera *et al.* [2017a], es una forma de referirse a problemas de usabilidad detectados en una IU o en la interacción del usuario con esta. La metodología implementada por la herramienta busca apuntar a una refactorización de usabilidad concreta para resolver los problemas de usabilidad relacionados con *usability smells*. Los *Usability Smells* considerados para el uso de la herramienta son: elemento no descriptivo, enlace erróneo, sin página de procesamiento, input libre para valores limitados, input sin formato, input corto, acción masiva innecesaria, contenido ignorado, contenido distante, sin validación con cliente, validación tardía, formulario no rellenado, resultados de búsqueda escasos, resultados de búsqueda inútiles, valor predeterminado incorrecto y elemento que no responde. Ante esto, es importante destacar que los *usability smells* son pistas o indicadores de problemas de usabilidad, por lo que en algunos casos no refieren directamente a un problema de usabilidad ya que depende del contexto asociado.

La estrategia utilizada para la identificación de los *usability smells* se basa en un proceso que consiste de tres fases: Registro de eventos, Detección de *usability smells* y reporte. Es importante mencionar que la arquitectura de este proceso se divide en un componente del lado del cliente que realiza el registro de eventos y un componente

del lado del servidor que es responsable de las dos fases restantes. En la Figura E.6 se puede apreciar un diagrama que expone esta secuencia de fases. En la fase de registro de eventos el componente del lado del cliente extrae eventos para filtrar y agregar los relevantes, siendo estos mencionados como eventos de usabilidad. En la fase de detección de *usability smells*, el componente del lado del servidor clasifica los eventos de usabilidad utilizando algoritmos especializados para descubrir los *usability smells*. Finalmente, el componente del lado del servidor también es responsable de la fase de reporte. Cuando los *usability smells* son identificados, se reportan junto con las refactorizaciones sugeridas que pueden resolverlos y con suficiente detalle como para que los desarrolladores y partes interesadas comprendan los problemas y tomen medidas para solucionarlos.

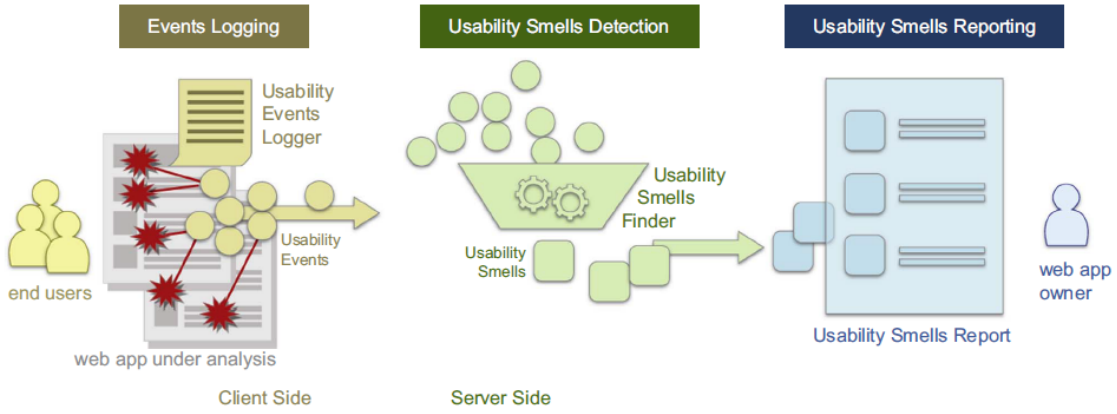


Figura E.6: Estrategia de identificación de *usability smells* y sus tres fases [Grigera *et al.*, 2017a].

Para la **fase de registro de eventos** se consideran ciertos eventos para poder realizar la detección de *usability smells*. Estos eventos son: intento de *tooltip*, intento de clic, desplazamiento rápido, navegación rápida, ruta de navegación, búsqueda, acción masiva (cuando se verifica un conjunto de elementos en una lista y luego se produce un envío), selección de opción, envío de formulario, formulario sin rellenar, input de texto y solicitud larga (cuando una solicitud tarda más que un umbral determinado). Cada uno de estos eventos se relaciona con uno de los *usability smells* nombrados anteriormente.

Para la **fase de detección de *usability smells*** se busca clasificar, agregar y analizar los eventos de usabilidad capturados. Los eventos descritos anteriormente se procesan en el momento en que llegan al servidor. La detección ocurre en tres

pasos: (i) Clasificación de eventos, donde se relaciona el evento detectado con su correspondiente *usability smell*; (ii) Síntesis de datos, donde tan pronto como el evento es clasificado, se extrae información clave de éste; y (iii) Evaluación de los *usability smells*, donde se reevalua la presencia de un *usability smell* en el elemento afectado, agregando nuevos *usability smells* si se detectan y manteniendo los ya asignados para que el informe esté completo. Se puede ver este proceso en la Figura E.7.

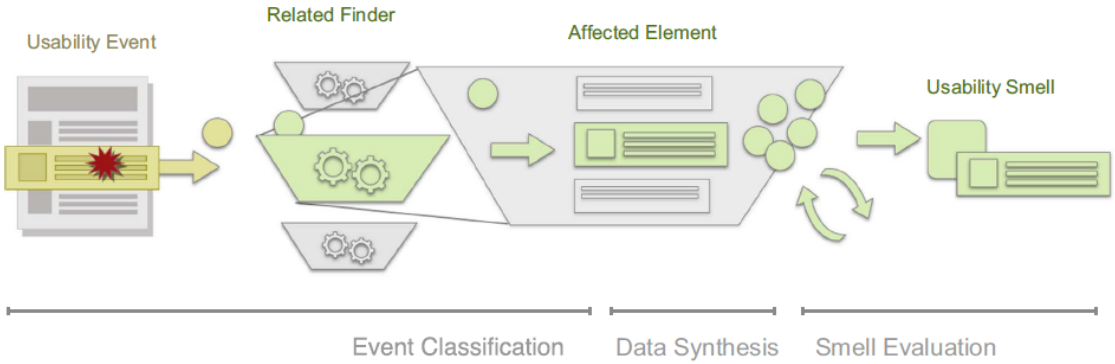


Figura E.7: Detección de *usability smells* [Grigera *et al.*, 2017a].

La **fase de reporte** consiste en informar los *usability smells* detectados junto con las refactorizaciones recomendadas que se puedan aplicar para poder solucionarlos. Cada *smell* se presenta con información destinada a ayudar al desarrollador a tomar una decisión.

Se implementa el enfoque explicado anteriormente con USF, una herramienta que funciona como SaaS. Los evaluadores de usabilidad pueden registrarse y la herramienta proporciona un fragmento de código que deben incrustar en la página web a evaluar. Esto permite que la herramienta registre los datos de la interacción para encontrar *usability smells*, que luego se informan gráficamente en la cuenta del propietario del sitio web.

La herramienta ofrece una mecánica de registro de eventos que considera la fase de registro de la interacción, analizando eventos de bajo nivel. Se procesan estos eventos y se filtran los que no ayuden a detectar *usability smells* para evitar tráfico innecesario de datos cuando estos sean enviados al lado del servidor. Los valores de umbrales para la generación de eventos de usabilidad pueden ser ajustados en el código inyectado en la web. Los eventos registrados no contienen información sobre los usuarios o sus dispositivos (como direcciones IP), tampoco registra contraseñas

o tarjetas de crédito, solo un texto auxiliar con el fin de indicar que se escribió algo. Destacar que el registro de interacción aborda eventos simples (los que se explicaron anteriormente) en lugar de tareas completas, por lo que no existe una forma de reconstruir la sesión de un usuario particular agrupando eventos.

La herramienta también ofrece mecánicas de detección de *smells*. Esto ocurre de lado del servidor ejecutando diferentes tipos de buscadores de *usability smells*. Cada uno detecta un *usability smell* específico. Si bien los parámetros de los buscadores vienen predeterminados, los evaluadores pueden configurarlos con base en sus necesidades. Los buscadores clasifican los *usability smells* de acuerdo con el elemento DOM afectado.

Por último, la herramienta describe mecánicas de reporte considerando la fase de reporte anteriormente explicada. La herramienta informa de los malos *usability smells* tal como aparecen. Para mostrar esto, Grigera *et al.* [2017a] explica que se desarrolló una interfaz web que muestra la información detallada del *usability smell*.

En la Figura E.8 se muestra una captura de pantalla de una aplicación real sometida a una evaluación utilizando USF. Se describe la etiqueta A como un *widget* que muestra los pasos que componen el proceso y el paso actual que se está ejecutando, pero no permite navegar hacia un paso anterior. La etiqueta B describe un mensaje de error que aparece después de presionar "enviar" y la validación falla.

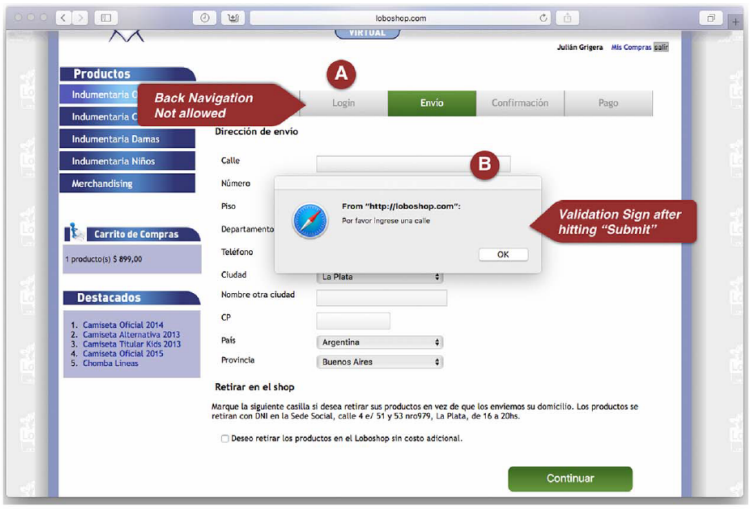


Figura E.8: Interfaz de aplicación sometida a evaluación [Grigera *et al.*, 2017a].

En la Figura E.9 se muestran dos capturas de pantalla de la herramienta USF que informan de dos *usability smells*: elemento que no responde (etiqueta A) y

validación tardía (etiqueta B). Para el primer caso, se detectó que muchos usuarios intentaron retroceder en el proceso de pago haciendo clic en los botones de los pasos anteriores sin darse cuenta que estos estaban atenuados. Para el segundo caso, se detectó, después de que muchos usuarios intentaron enviar un formulario incompleto, que no indicaba los campos faltantes obligatorios hasta que presionaron "enviar". Para cada caso se muestran datos específicos. Para el primer caso, se muestra el número de veces que se hizo clic en el elemento y para el segundo caso se muestra un porcentaje de envíos de formularios no satisfactorios. La herramienta también sugiere refactorizaciones para los *usability smells* detectados.

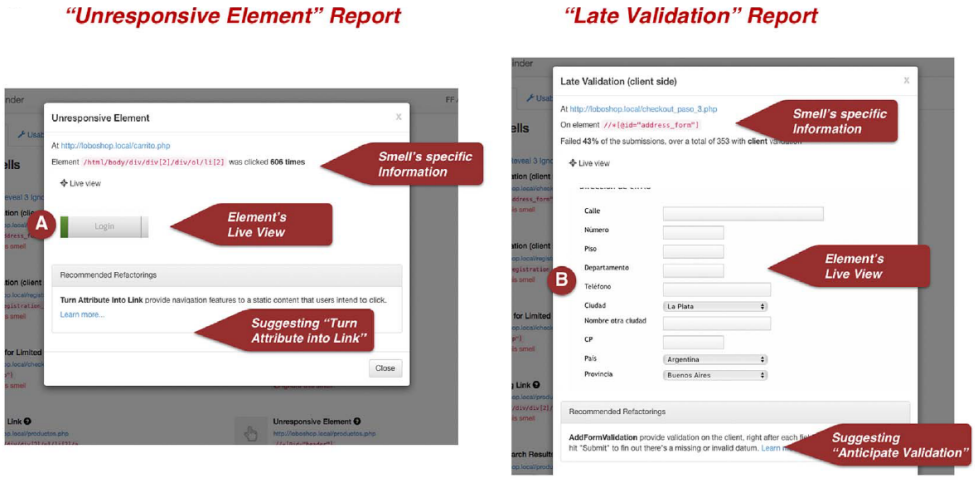


Figura E.9: Interfaz de USF indicando los *usability smells* detectados y las refactorizaciones recomendadas [Grigera *et al.*, 2017a].

En la Figura E.10 se muestra que los evaluadores también pueden ignorar *usability smells*, ya sea porque no consideran ese *smell* o por un falso positivo. Además, en la Figura E.10 se aprecia como USF muestra la vista general de los *usability smell* detectados, mostrando las pestañas que permiten gestionar las refactorizaciones recomendadas y los eventos detectados relacionados con estos *usability smells*. Con esto, el evaluador tiene las herramientas necesarias para poder gestionar una correcta corrección de los problemas de usabilidad detectados.

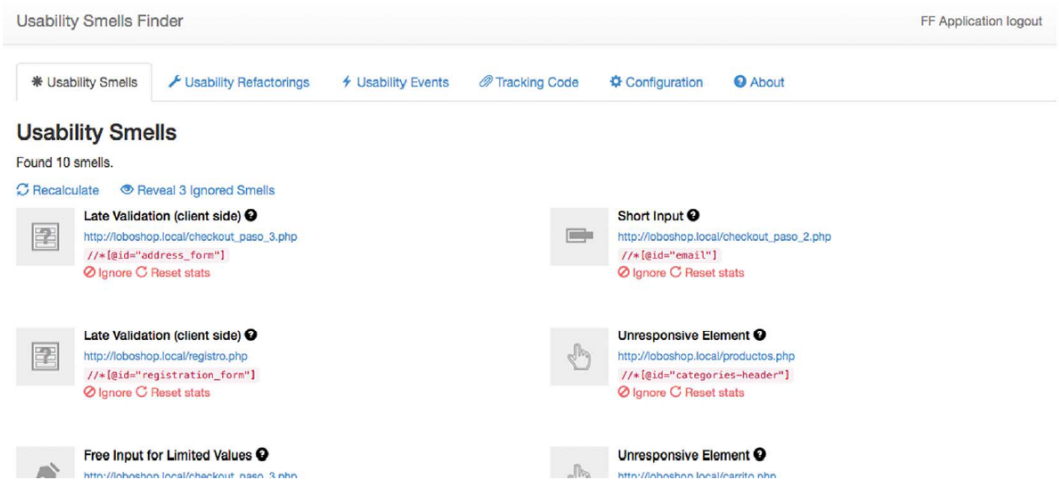


Figura E.10: Vista general de los *usability smells* detectados por USF [Grigera *et al.*, 2017a].

E.4. MUSE (*Mobile Usability Smell Evaluation*)

MUSE es una herramienta que apoya la evaluación automática de la usabilidad detectando problemas de usabilidad en páginas web enfocadas a entornos móviles y de escritorio. Está basada en proxy y permite registrar el comportamiento de un usuario mientras interactúa con cualquier aplicación web a través de dispositivos de escritorio o móviles, utilizando estos datos de interacción para determinar *bad usability smells* [Paternò *et al.*, 2017]. Esta herramienta pertenece a la categoría de Herramientas que detectan problemas de usabilidad.

Los datos de interacción son registrados con JavaScript, que es inyectado en la página web a través de un servidor proxy, por lo que no es necesario que el propietario del sitio tenga que hacerlo manualmente. En la Figura E.11 se puede apreciar la arquitectura de MUSE.

El proxy también ofrece la opción de indicar tareas a realizar en esa aplicación por parte de los usuarios al inicio de la sesión de pruebas de usabilidad. Las interacciones de los usuarios se registran como una secuencia de eventos. MUSE es capaz de detectar eventos de interacción tanto de entornos de escritorio (eventos relacionados al uso del mouse y teclado) como de sitios web móviles (eventos relacionados a pantallas *touch*, así como de los sensores del *smartphone*). Las pruebas de usabilidad pueden ser creadas y eliminadas, además de que pueden ser indicadas las tareas que las componen.

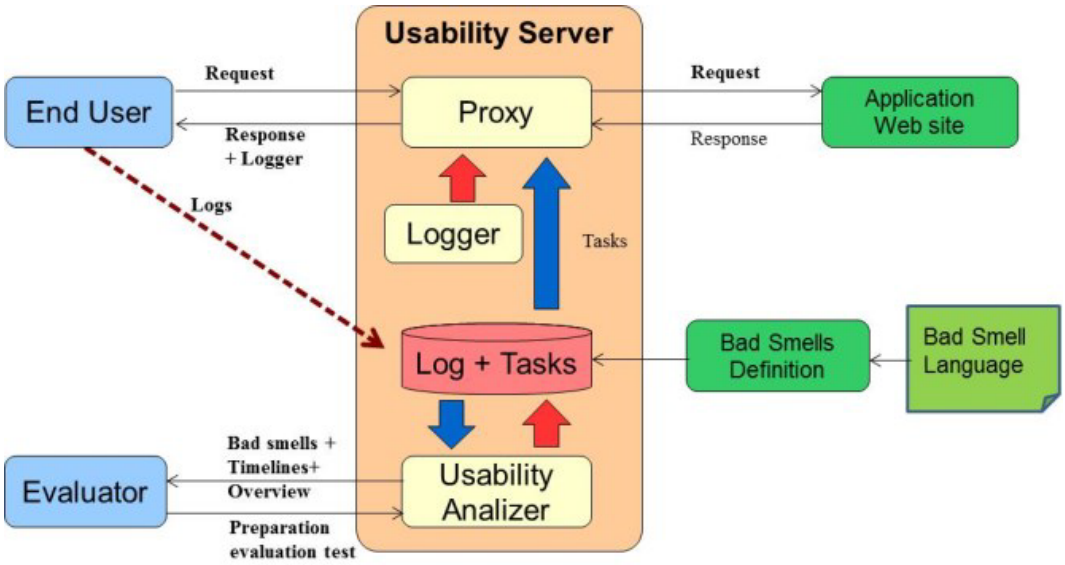


Figura E.11: Arquitectura de MUSE [Paternò *et al.*, 2017].

La herramienta incluye un módulo analizador de usabilidad que toma los datos recopilados durante las pruebas de usabilidad y proporciona información general de los registros recopilados y visualizaciones de la línea de tiempo interactiva asociada, además de indicaciones de dónde se han producido los *bad usability smells* en dichos registros. Las líneas de tiempo, que se derivan de diferentes registros de usuario, pueden superponerse para comparar los distintos comportamientos de estos ante tareas definidas. Para cada evento detectado, se registra el tipo de evento (y si es significativo), se marca: el tiempo del evento, la etiqueta HTML en la que se activó el evento, las coordenadas del punto de la pantalla donde se ha disparado el evento y otras informaciones, dependiendo del tipo de evento (como caracteres escritos, coordenadas GPS, ruta de la captura de pantalla de la página, etc.). Además, los evaluadores pueden definir algunos eventos personalizados.

MUSE basa su funcionamiento en la detección de *bad usability smells*, lo cual es definido por Paternò *et al.* [2017] como potenciales problemas de usabilidad. Paternò *et al.* [2017] destacan que consideraron problemas de usabilidad para ser identificados por MUSE, tales como elementos demasiado pequeños o cercanos, hipervínculos demasiado cercanos, contenido demasiado alejado, secciones demasiado pequeñas, mala legibilidad y formularios demasiado extensos. De acuerdo con estos problemas de usabilidad, la herramienta MUSE realizará la búsqueda de *bad usability smells*.

La detección de *bad usability smells* considerando las interacciones registradas por los usuarios se realiza utilizando el siguiente algoritmo: seleccionar todos los eventos en los registros que pueden ser el comienzo de una de las subsecuencias de interés, y de cada uno de ellos comenzar a construir una subsecuencia candidata, verificando paso a paso la consistencia de la descripción del patrón generado. Si una subsecuencia candidata es incompatible con la descripción del patrón, entonces se elimina, reduciendo así el conjunto de subsecuencias candidatas. Es por esto que el algoritmo depende de dos condiciones previas: la descripción de este patrón de subsecuencias y que éste debe comenzar siempre con un evento de un tipo específico; y cada registro debe estar asociado con un índice numérico cronológicamente progresivo. Este algoritmo de detección de *bad usability smells* se puede aplicar a cada registro inmediatamente después de terminada la sesión de pruebas de usabilidad.

En la Figura E.12 se puede apreciar la línea de tiempo generada por MUSE, correspondiente a la secuencia de acciones creada por el usuario al realizar las pruebas de usabilidad. La interacción de éste se refleja en esta línea de tiempo, permitiendo a los evaluadores ver de forma gráfica donde se encuentran los *bad usability smells*. Cada ícono muestra el tipo de evento. En esta misma línea de tiempo se marca en rojo el *bad usability smell* detectado, indicando una secuencia de acciones que MUSE detecta como problema de usabilidad. Cada evento tiene asociado una captura de pantalla, lo que permite a los evaluadores y desarrolladores tener una mejor comprensión de la IU.

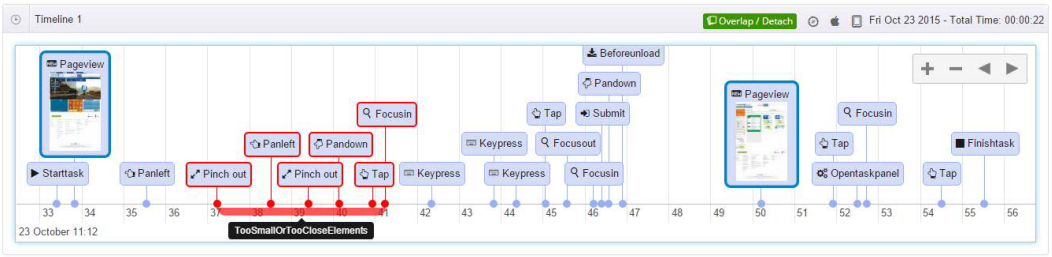


Figura E.12: Línea de tiempo de interacción de usuario en MUSE. Se destaca en rojo un *bad usability smell* [Paternò *et al.*, 2017].

E.5. Kobold

Kobold es una herramienta que apoya la evaluación automática de la usabilidad detectando problemas de usabilidad en páginas web enfocadas a entornos de escritorio [Grigera *et al.*, 2017b]. Kobold está construida teniendo como base la herramienta USF [Grigera *et al.*, 2017a], por lo que su funcionamiento es bastante parecido. Se agregan funciones adicionales que brindan mayor funcionalidad y utilidad en materia de evaluación automática de la usabilidad. Esta herramienta pertenece a la categoría de Herramientas que corrigen problemas de usabilidad.

Kobold es la herramienta que implementa el proceso completo de automatizar la mejora de la usabilidad web. Funciona como SaaS por lo que se comporta como una aplicación web en sí misma. Las principales características que ofrece esta herramienta son la detección de problemas de usabilidad manifestados como *usability smells* con detalles específicos, la sugerencia de soluciones a los *usability smells* detectados en términos de refactorizaciones de usabilidad, la aplicación automática de las refactorizaciones cuando sea posible y un reporte instantáneo, además de permitir navegar a través de todos los eventos de usabilidad detectados. Se destaca que con esta herramienta cualquiera que quiera realizar una evaluación de la usabilidad, ya sea novato o experto en el tema, pueda hacer uso de la detección y refactorización de problemas de usabilidad.

Puesto que Kobold está basada en USF [Grigera *et al.*, 2017a], hereda funcionalidades de esta. Entre estas funcionalidades se destacan las tres etapas del proceso de detección de problemas de usabilidad. La primera consiste en detectar todos los eventos de la interfaz del usuario en el cliente y recopilar los relevantes. En la segunda etapa, los eventos de usabilidad se analizan en un servidor externo para detectar los *usability smells*. Por último, en la tercera etapa se realizan las recomendaciones de refactorizaciones de usabilidad para solucionar dichos problemas. En algunos casos, estas refactorizaciones se pueden aplicar de forma automática o semiautomática con algunas aportaciones de los desarrolladores. Esta última etapa corresponde a la herramienta presentada de nombre Kobold [Grigera *et al.*, 2017b], complementando los aspectos faltantes que USF no cubría.

Para su uso, los desarrolladores deben registrarse. Después de esto, la herramienta proporcionará un fragmento de código JavaScript para incrustar en el encabezado de la aplicación web. Con esto, la herramienta inmediatamente comienza a registrar la interacción de los usuarios finales en búsqueda de *usability smells*. Cuando se

reporta un problema de usabilidad, Kobold sugiere una o más refactorizaciones para solucionarlo. Dependiendo del caso, la refactorización se puede presentar como una sugerencia (que debe ser implementada por el desarrollador) o se puede ofrecer como una solución automatizada. Para algunas refactorizaciones la herramienta requerirá de información adicional del desarrollador antes de ser aplicada automáticamente (estas se denominan según Grigera *et al.* [2017b] como refactorizaciones semiautomáticas).

En la Figura E.13 se muestra una captura de pantalla de Kobold en donde se sugiere una refactorización mencionada como *Add Autocomplete* para corregir el *usability smell* "Input libre para valores limitados". La captura muestra el diagnóstico del *usability smell* con una vista del elemento afectado (en este caso, el input "*Country*") y solicita al usuario que confirme la aplicación de la refactorización o edite los valores para las sugerencias de autocompletado.

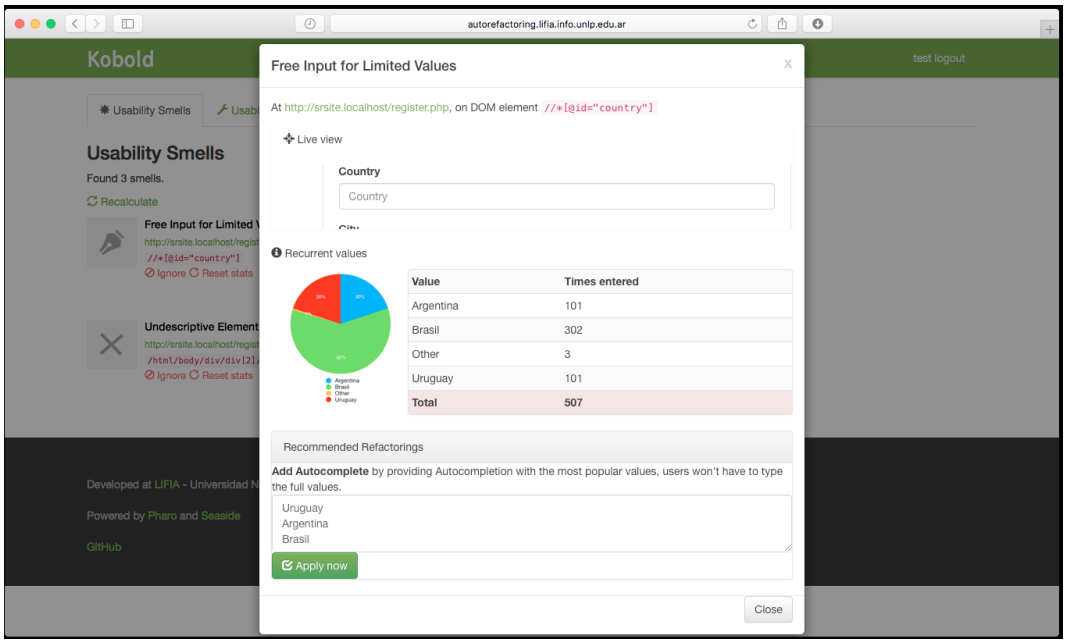


Figura E.13: Captura de pantalla de Kobold. Se muestra una ventana emergente indicando la sugerencia de refactorización [Grigera *et al.*, 2017b].

Ante el caso de que no siempre se realicen refactorizaciones automáticas y conociendo que Kobold puede entregar más de una recomendación de refactorización (e incluso elegir la mejor recomendación entre todas las dadas), Grigera *et al.* [2017b] clasifica tres grupos de estas refactorizaciones: (i) Solo sugerencias, donde la herramienta solo entrega recomendaciones para que estas sean implementadas por

un desarrollador, (ii) Semiautomatizadas, donde la herramienta puede generar automáticamente refactorizaciones, pero requieren algunos datos de parámetros que un desarrollador debe proporcionar manualmente y (iii) Totalmente automatizados, donde el *usability smell* detectado proporciona todos los datos necesarios para que Kobold sea capaz de aplicar la refactorización automáticamente sin ayuda externa.

Como Kobold fue construida con base en USF, su arquitectura es similar, por lo que captura eventos de la IU de la interacción de usuarios reales del lado del cliente, extrayendo y filtrando los eventos de interés, además de agregar y clasificar los eventos del lado del servidor para descubrir *usability smells*. Dada la naturaleza inmediata del análisis de registros, la herramienta informa los *usability smells* que puedan afectar a la aplicación web analizada en el momento en el que aparecen. La tecnología que permite aplicar refactorizaciones de forma automática es posible gracias a la implementación de un *framework* de adaptación del lado del cliente que adapta las aplicaciones existentes cambiando su DOM. Cuando las refactorizaciones son automáticas o semiautomáticas, Kobold genera automáticamente el código JavaScript a partir de una plantilla fija. Si las refactorizaciones se aplican a un elemento DOM, la plantilla se completa dinámicamente localizando el elemento a refactorizar.

Utilizando las técnicas previamente mencionadas, Kobold puede ofrecer una aplicación de refactorización de usabilidad de forma automática o semiautomática. Esta es la única herramienta encontrada en la investigación presentada en este trabajo que permite implementar correcciones de problemas de usabilidad de forma automática, por lo que es de gran valor a la hora de querer implementar una evaluación de la usabilidad de forma automática, específicamente, para interfaces de usuario de aplicaciones web en entornos de escritorio.

E.6. Plain

Plain es un plug-in de Eclipse que apoya la evaluación automática de la usabilidad detectando problemas de usabilidad en interfaces de usuario móviles mediante la comparación de métricas de los elementos que componen la interfaz [Soui *et al.*, 2017]. Esta herramienta pertenece a la categoría de Herramientas que detectan problemas de usabilidad.

Plain permite predecir la usabilidad de una IU móvil detectando problemas de calidad de los elementos que conforman la interfaz en función de métricas de usabilidad. Toma como input un proyecto Java correspondiente a la aplicación móvil que se desea evaluar, generando una lista de los problemas detectados como resultado. Las métricas utilizadas corresponden a un set de métricas de usabilidad que se pueden clasificar según dos criterios: (i) Orientación (dentro de ésta se encuentran aspectos como regularidad, composición, clasificación y complejidad) y (ii) Coherencia (dentro de ésta se encuentran aspectos como integridad, densidad, repartición y simetría).

En la Figura E.14 se puede apreciar la arquitectura de Plain. Esta incluye tres módulos: (i) Extractor de propiedades de la IU móvil, (ii) Calculadora de métricas de evaluación y (iii) Ajuste de métricas. Al proporcionar la aplicación de entorno móvil, Plain obtiene todas las propiedades de los componentes de la IU móvil. Luego, estas propiedades se utilizan para calcular las métricas de evaluación de calidad. Con estas propiedades se ajustan las métricas de evaluación. Por último, Plain genera la lista de defectos correspondientes a la IU móvil que se está evaluando. Para poder utilizar Plain, el evaluador debe importar el código fuente de la IU móvil del proyecto a evaluar y abrir la *navigator view* del editor de Eclipse.

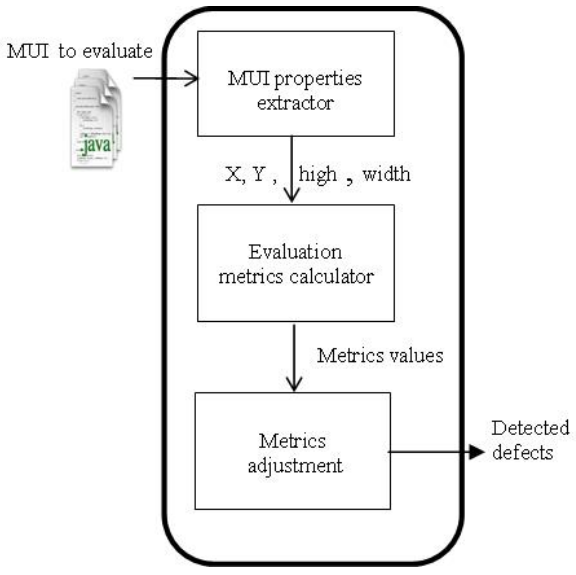


Figura E.14: Arquitectura de Plain [Soui *et al.*, 2017].

El primer módulo de Plain, correspondiente a la extracción de propiedades de la IU móvil, permite extraer dichas propiedades para luego medirlas y compararlas con

las métricas de usabilidad. Plain extrae el ancho, alto, alineación, eje de coordenadas para el punto izquierdo superior de un objeto. El extractor analiza el código fuente de la IU móvil y extrae las propiedades.

El segundo módulo de Plain, que es la calculadora de métricas de evaluación, tiene como objetivo calcular los valores de las propiedades obtenidas del módulo anterior y calcular las métricas de calidad. Este módulo entrega como resultado la medida de estas métricas de calidad. En la Figura E.15 se puede apreciar como la herramienta Plain permite mostrar estas métricas de evaluación.

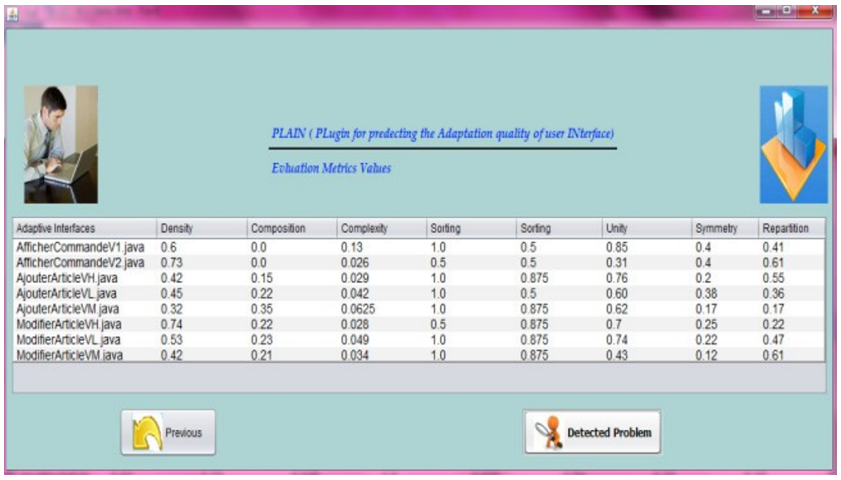
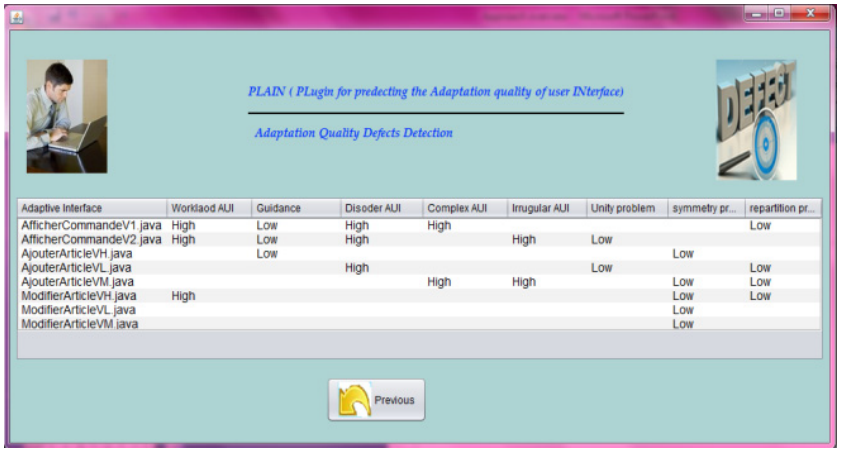


Figura E.15: Captura de pantalla del módulo de calculadora de métricas de evaluación de Plain, mostrando el valor de las métricas de los elementos de la IU móvil [Soui *et al.*, 2017].

El tercer módulo de Plain, que es el ajuste de métricas, realiza un ajuste de los resultados obtenidos en el módulo anterior. Esto es debido a que es difícil generalizar los umbrales que se deben considerar a la hora de evaluar la IU móvil ya que son diferentes en cuanto a número de interfaces por aplicación, número de componentes por interfaz móvil, etc. Con las métricas ajustadas, se puede realizar la detección de problemas de usabilidad.

La detección de problemas de usabilidad busca comparar los datos entregados por los módulos explicados anteriormente con los umbrales definidos por el autor [Soui *et al.*, 2017]. Sin embargo, Plain también permite a los desarrolladores y evaluadores definir y especificar sus propios umbrales. Una vez los valores de los umbrales se adaptan a las aplicaciones móviles actuales, se pueden detectar problemas de usabilidad. Los problemas se detectan una vez que los valores de las métricas

de los elementos de la IU móvil no se encuentran dentro de los valores permitidos por el umbral definido. En la Figura E.16 se observa como Plain despliega la lista de problemas de usabilidad detectados en la IU móvil evaluada. Además, se puede apreciar la valoración que tienen estos problemas refiriéndose a las métricas consideradas. Plain considera, de acuerdo con lo explicado anteriormente, métricas de regularidad, composición, clasificación, complejidad, integridad y densidad.



Adaptive Interface	Workload AUI	Guidance	Disoder AUI	Complex AUI	Irrugular AUI	Unity problem	symmetry pr...	repartition pr...
AfficherCommandeV1.java	High	Low	High	High				Low
AfficherCommandeV2.java	High	Low	High		High	Low		
AjouterArticleVH.java		Low					Low	
AjouterArticleVL.java			High			Low		Low
AjouterArticleVM.java				High	High		Low	Low
ModifierArticleVH.java	High						Low	Low
ModifierArticleVL.java							Low	
ModifierArticleVM.java							Low	

Figura E.16: Captura de pantalla de Plain mostrando la lista de los problemas de usabilidad detectados [Soui *et al.*, 2017].

E.7. UTAssistant

Esta herramienta corresponde a los estudios primarios de los autores Desolda *et al.* [2017], Federici *et al.* [2018] y Federici *et al.* [2019]. UTAssistant surge de la necesidad de mejorar la forma en la que se hacían pruebas de usabilidad remotas. Esta herramienta utiliza HTML y JavaScript para solucionar esta necesidad. Se presenta como una herramienta simple que permite fomentar la participación de los usuarios finales a la hora de realizar pruebas de usabilidad de sitios web. UTAssistant registra datos del computador del usuario como la interacción de este con el teclado y mouse, además de permitir gestionar cuestionarios y grabar video y audio. Esta herramienta pertenece a la categoría de Herramientas que apoyan la evaluación de la usabilidad.

Los evaluadores pueden crear y gestionar pruebas de usabilidad con la herramienta UTAssistant. Una prueba de usabilidad parte con el diseño de la prueba, que consiste en (i) crear un script para introducir a los usuarios a la prueba, (ii) definir

un conjunto de tareas a realizar por el usuario final, (iii) identificar los datos que se recopilarán (como clics, tiempo requerido para realización de tarea, grabación de video/audio, etc.) y (iv) decidir que cuestionarios administrar a los usuarios. El asistente de UTAssistant permite a los evaluadores realizar estas actividades guiando el diseño de una prueba, ayudando a los evaluadores a crear listas de tareas y enviando a los usuarios el enlace por correo electrónico para que estos realicen las pruebas de usabilidad.

Después de diseñar la prueba de usabilidad, los usuarios reciben un correo electrónico con información sobre la evaluación que se les pide completen, junto con un enlace para acceder a UTAssitant. Seguido de esto, los usuarios pueden realizar la prueba de usabilidad. La ejecución de cada tarea es guiada por la herramienta, que muestra la descripción de la tarea en una ventana emergente y luego abre la página web en la que se debe realizar la tarea. En la Figura E.17 se muestra la barra de herramientas, ubicada en la parte superior de la página web, donde se agrupan las funciones de UTAssistant. Esta barra indica el título de la tarea actual, su objetivo, la duración de la tarea, el número de tarea y un botón para pasar a la siguiente tarea. Durante la ejecución de la tarea, la herramienta recopila todos los datos identificados por el evaluador.

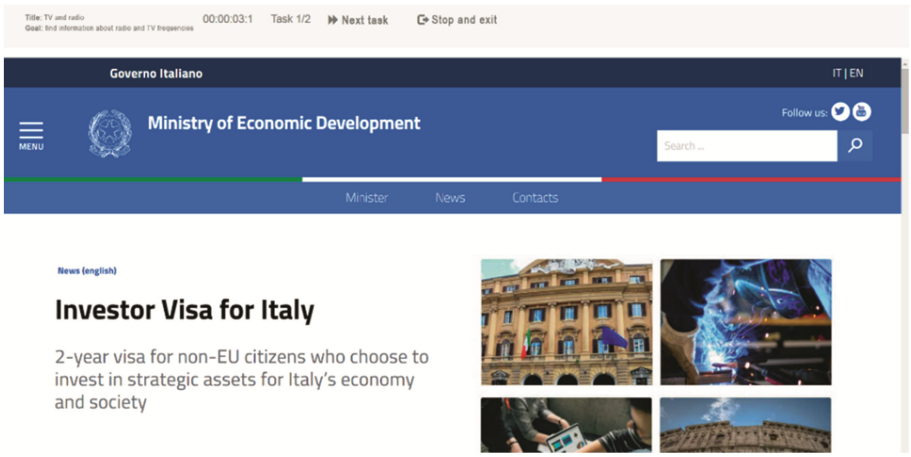


Figura E.17: Un ejemplo de ejecución de tarea. La barra de tareas de UTAssistant se muestra sobre la página web a evaluar [Federici *et al.*, 2018].

UTAssistant automatiza las actividades de registro de interacción con mouse y teclado, la grabación de video y audio y la gestión del cuestionario. Para el caso de los cuestionarios, UTAssistant almacena automáticamente las respuestas de los usuarios y reporta resultados en forma de estadísticas y gráficos. La herramienta

permite grabar: la voz del usuario con el micrófono, sus expresiones faciales con la webcam y la pantalla del escritorio. Este contenido registrado puede ser analizado por los evaluadores para comprender los problemas de usabilidad que se puedan detectar. Para que esta funcionalidad sea más efectiva, UTAssistant proporciona herramientas de notas, pudiendo crear acotaciones en los videos y audios grabados. Por último, UTAssistant rastrea el comportamiento del usuario mediante la recopilación de registros de mouse y teclado. Con base en estos datos, la herramienta proporciona al evaluador estadísticas de rendimiento para cada tarea.

E.8. Guideliner

Guideliner es una herramienta que evalúa la conformidad de las interfaces de usuario web con pautas de usabilidad durante la fase de implementación del desarrollo de una interfaz [Marenkov *et al.*, 2018]. Contiene un conjunto predefinido de pautas de usabilidad y permite definir pautas de usabilidad personalizadas. Esta herramienta pertenece a la categoría de Herramientas que detectan problemas de usabilidad.

Guideliner ayuda a los desarrolladores a detectar problemas de usabilidad después de cada modificación en la IU, lo que permite una retroalimentación inmediata sobre estos problemas. Esta herramienta está organizada como un proyecto conformado por varios módulos de Java. La Figura E.18 muestra la arquitectura de Guideliner, mostrando sus componentes. Guideliner se divide en dos componentes: la implementación (oculta para los usuarios) y la interfaz de programación pública que proporciona interfaces para iniciar el proceso de evaluación de la usabilidad.

Una vez que el evaluador de usabilidad incorpora los componentes necesarios para la evaluación automatizada de la usabilidad (componentes mostrados en la Figura E.18), Guideliner carga todas las pautas de usabilidad (definidas por los autores en su artículo [Marenkov *et al.*, 2018]). El motor de evaluación de la usabilidad procesa la IU web y evalúa los elementos de la interfaz con respecto a las métricas definidas en las pautas de usabilidad.

El repositorio de pautas contiene pautas de usabilidad adecuadas para la evaluación. Dentro de esta funcionalidad, los elementos de la IU web son comparados con los valores que definen las pautas, calculando previamente las propiedades de dichos elementos.

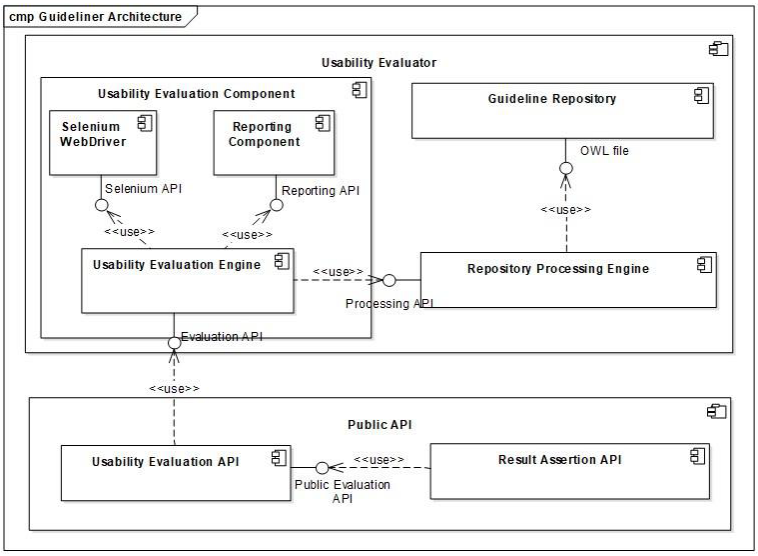


Figura E.18: Diagrama de componentes que muestra todos los elementos de la herramienta Guideliner y sus relaciones [Marenkov *et al.*, 2018].

El motor de usabilidad se basa en *Selenium WebDriver*, que proporciona operaciones y comandos para buscar la página a analizar, ubicar los elementos de la IU, hacer clic en elementos de la interfaz, completar entradas, moverse entre ventanas, etc. Los valores reales de las características del elemento de la IU son guardados como parámetros en adaptadores de evaluación, correspondientes a cada tipo de elemento de la interfaz. Los valores de estos adaptadores son evaluados mediante *Selenium WebDriver* y afirma si los valores recuperados corresponden al valor definido en la pauta de usabilidad. Las pautas de usabilidad usadas por la herramienta Guideliner son rescatadas de recomendaciones de publicaciones científicas, WCAG, pautas introducidas por el Departamento de Salud y Servicios Humanos de la U.S. En total se dispone de 98 pautas que se pueden procesar automáticamente.

En la Figura E.19 se pueden apreciar los resultados de una evaluación de la usabilidad luego de ejecutar pruebas para verificar la conformidad de la IU web con las pautas de usabilidad. El círculo verde muestra que la prueba pasó, mientras que un círculo naranja resalta las pruebas fallidas. En caso de que las pruebas fallen, se proporciona información adicional sobre la infracción, incluido el tipo de elemento, el texto del elemento y el motivo de la falla.

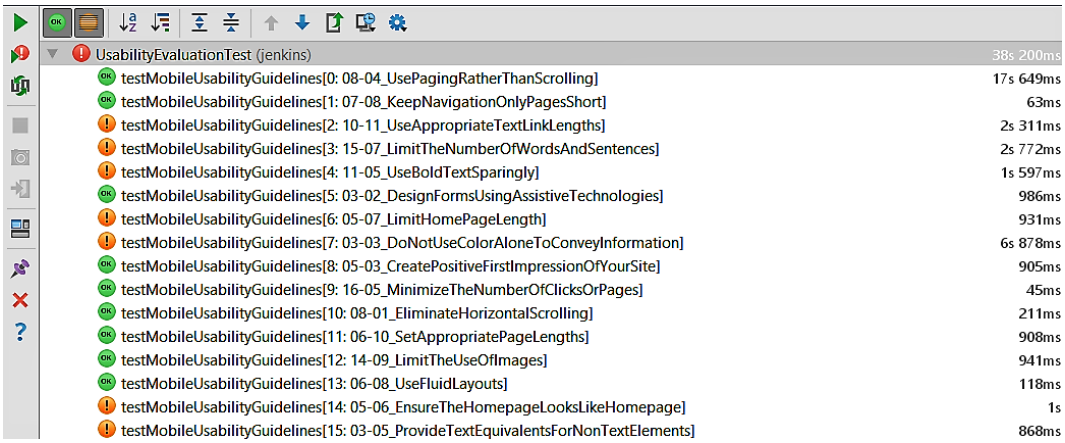


Figura E.19: Resultados de la evaluación automática de usabilidad utilizando la herramienta Guideliner [Marenkov *et al.*, 2018].

E.9. I2Evaluator

I2Evaluator (también conocida como *Interactive Iterative Evaluator*) es una herramienta que apoya la evaluación de la usabilidad de aplicaciones web en entornos de escritorio que se basa en medir la usabilidad de interfaces de usuario adaptables utilizando métricas estéticas [Chettaoui y Bouhlel, 2017]. Esta herramienta pertenece a la categoría de Herramientas que miden la usabilidad.

Estas métricas se desprenden de estándares de calidad (como ISO 25010 para la calidad del software [ISO, 2011]). I2Evaluator considera un conjunto de indicadores, entre los que se encuentran la funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad, además de incluir factores como el rendimiento y ergonomía estética. Esta última se destaca por la importancia que conlleva los juicios estéticos de una IU a la hora de realizar una evaluación de la usabilidad. I2Evaluator surge de la necesidad de considerar esta variable, por lo que se basa en métricas estéticas para obtener valores comparables de diferentes interfaces de usuario adaptables.

I2Evaluator está implementada en un servicio web generado con AngularJS y su interfaz consta de dos pantallas principales. La primera se muestra cuando se conecta por primera vez la página web y contiene dos áreas donde el evaluador debe soltar una o dos capturas de pantalla de la IU que requiere evaluar para proceder con el funcionamiento de la herramienta. En la Figura E.20 se puede apreciar esta

pantalla principal. Un botón de *Continue* da el pase a la segunda pantalla de la herramienta, una vez se han subido las capturas de pantalla.

La segunda página es la parte principal de la herramienta donde se pueden evaluar las interfaces adaptativas. Primero se debe especificar el tamaño de la ventana en la que se desea implementar la interfaz. Esto se puede apreciar en la Figura E.21 en donde, a modo de ejemplo, el evaluador entrega un valor de 30 para el tamaño de ventana. En esta segunda pantalla, la herramienta manipula las capturas de pantalla mediante un algoritmo definido de descomposición de imágenes.

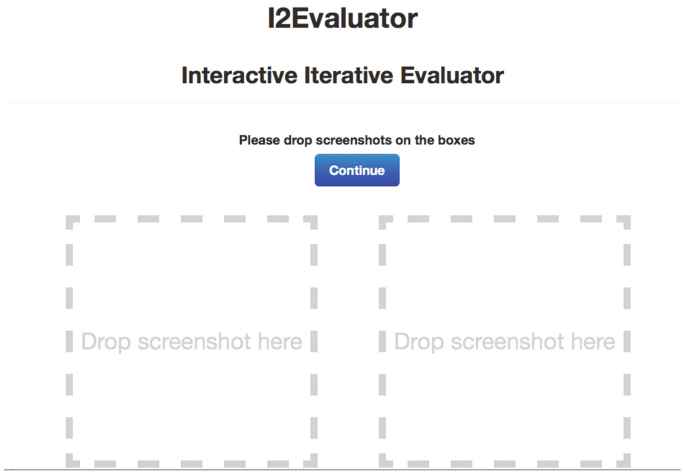


Figura E.20: Pantalla principal de la herramienta I2Evaluator [Chettaoui y Bouhlel, 2017].

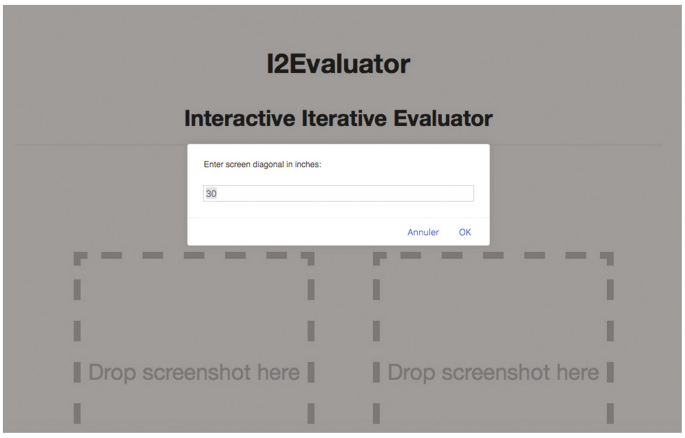


Figura E.21: Segunda pantalla de la herramienta I2Evaluator. La herramienta solicita al evaluador que especifique el tamaño de la ventana de la plataforma objetivo [Chettaoui y Bouhlel, 2017].

I2Evaluator ofrece una pestaña de métricas al extremo derecho de cada interfaz que muestra algunas medidas de métricas estéticas. En la Figura E.22 se puede apreciar esto, mostrando un cuadro con las medidas de las métricas obtenidas gracias al análisis realizado por la herramienta. Entre estas métricas se pueden observar el balance, la densidad, la complejidad y el alineamiento de las interfaces subidas por el evaluador.

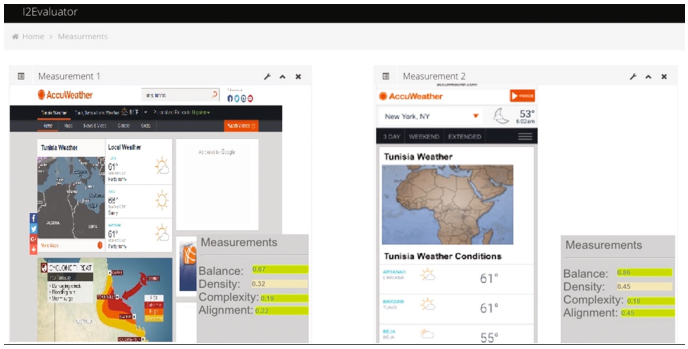


Figura E.22: Mediciones de métricas estéticas realizadas por la herramienta I2Evaluator [Chettaoui y Bouhlel, 2017].

El principal objetivo de I2Evaluator es proporcionar a los diseñadores un análisis objetivo de la IU final evaluada, entregando una medición de los valores considerados para la evaluación de la usabilidad.

E.10. PlatoS

PlatoS es una herramienta que apoya la evaluación de la usabilidad de *mockups*. Permite al evaluador integrar los *mockups* de Android y capturar un caso de uso en términos de una secuencia de acciones, directamente en el dispositivo móvil [Barra *et al.*, 2019]. Esta herramienta pertenece a la categoría de Herramientas que detectan problemas de usabilidad.

PlatoS presenta una arquitectura cliente-servidor y permite la identificación automática de problemas de usabilidad a través de un análisis estadístico de los datos registrados. En la Figura E.23 se muestra la arquitectura de PlatoS, detallando sus componentes.

PlatoS almacena los *mockups* que necesitan ser evaluados. Para que la herramienta pueda detectar problemas de usabilidad primero se debe simular el uso del

mockup. Para esto, el evaluador debe descargar estos *mockups* usando PlatoS directamente desde el dispositivo móvil. Para cada IU, el evaluador simula el uso de la misma mientras que la herramienta registra los datos de secuencia de acciones y toma los tiempos de cada acción. Los datos recopilados se transfieren luego al servidor de PlatoS. En la Figura E.24 se puede apreciar la interfaz de PlatoS del lado del desarrollador siendo utilizada en un dispositivo móvil. La pantalla izquierda muestra las opciones para *Importar interfaz* y *Manejar secuencia*. La pantalla derecha muestra como PlatoS permite gestionar las secuencias de acciones para las pruebas de usabilidad de cada *mockup*.

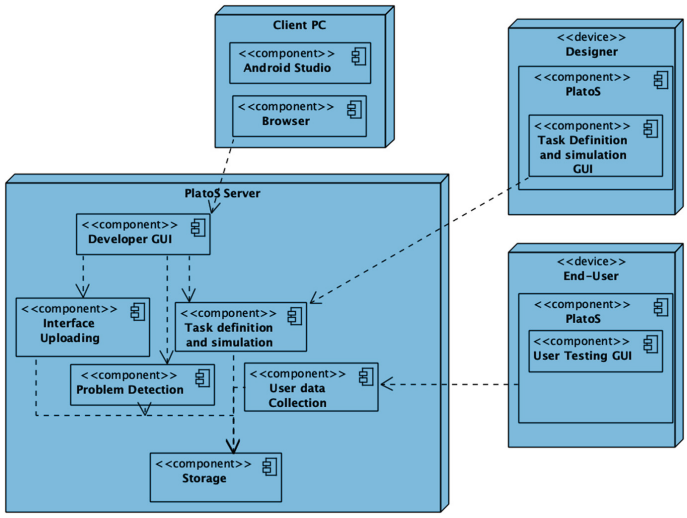


Figura E.23: Arquitectura de la herramienta PlatoS [Barra *et al.*, 2019].

Los usuarios finales descargan del servidor las interfaces de usuario correspondientes a los *mockups*, además de las tareas a realizar y los tiempos de la interacción ejecutada por el desarrollador. Luego, simulan el uso de la IU realizando las actividades correspondientes. Los comentarios de los usuarios finales se recopilan después de cada prueba y se transfieren al servidor.

Para la detección de problemas, PlatoS dispone de un conjunto de métricas de usabilidad, las cuáles se comparan automáticamente con las del desarrollador a través de un análisis estadístico y se crea un informe detallado, accesible para el diseñador a través de la web. En caso de que la interfaz tenga algún problema destacado, el diseñador analiza el problema e intenta solucionarlo modificando adecuadamente el *mockup*.

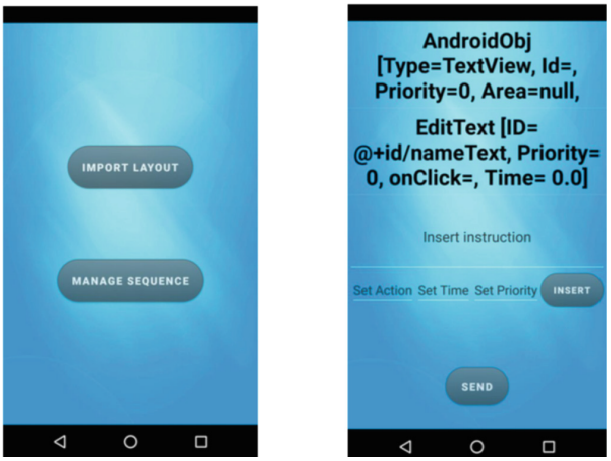


Figura E.24: Interfaz de la herramienta PlatoS del lado del desarrollador [Barra *et al.*, 2019].

Una vez realizadas las pruebas de usabilidad por parte de los usuarios finales, PlatoS permite explorar los resultados utilizando un tablero analítico. En la Figura E.25 se muestra la comparación de tiempo óptimo de una tarea con respecto a los tiempos de los usuarios que realizaron la misma tarea.

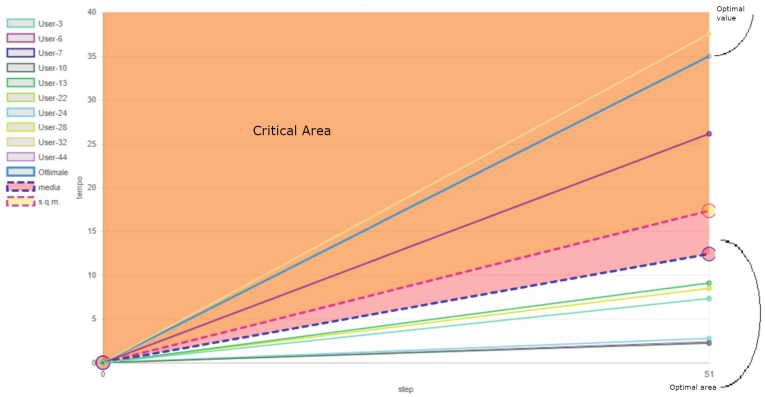


Figura E.25: Un ejemplo de panel de comparación de PlatoS sobre una tarea específica [Barra *et al.*, 2019].

En la Figura E.26 se muestran los tiempos de una tarea de prueba compuesta por siete pasos. Cada línea se refiere a un usuario que realizó las pruebas de usabilidad. En esta figura se puede apreciar con mejor detalle el comportamiento de los usuarios con respecto a cada paso a realizar en la tarea, por lo que se puede tener una vista más detallada de donde se pueden encontrar los problemas de usabilidad específicamente.

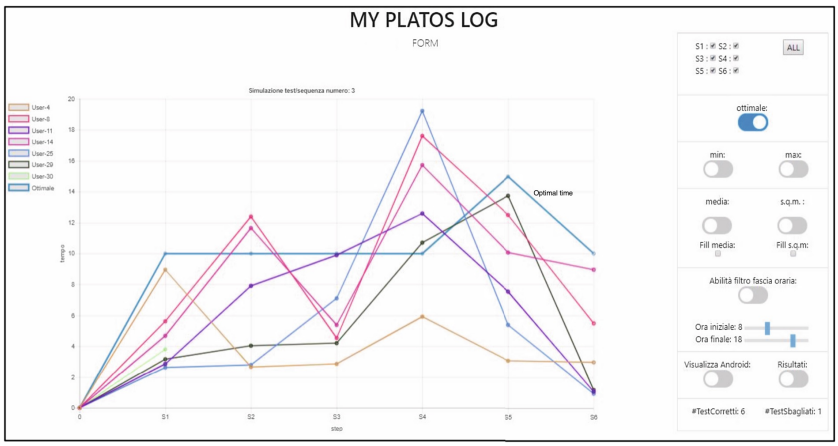


Figura E.26: Un ejemplo de panel de comparación de PlatoS sobre una tarea conformada por siete pasos [Barra *et al.*, 2019].

E.11. OwlEye

OwlEye es una herramienta que apoya la evaluación de la usabilidad de aplicaciones móviles. Esta herramienta, se basa en tecnologías como el *deep learning* para detectar problemas de usabilidad de capturas de pantalla de interfaces de usuario de aplicaciones móviles [Liu *et al.*, 2020]. Esta herramienta pertenece a la categoría de Herramientas que detectan problemas de usabilidad.

Inspirándose en el hecho de que los ojos humanos pueden detectar fácilmente los errores de visualización, Liu *et al.* [2020] propone la herramienta OwlEye. Esta herramienta permite modelar la información visual mediante *deep learning* para detectar y localizar automáticamente los problemas de visualización de la IU. OwlEye está basada en una CNN para identificar en las capturas de pantalla los problemas de visualización y utiliza un Grad-CAM (*Gradient weighted Class Activation Mapping*) para localizar las regiones con problemas de usabilidad en la IU para guiar a los desarrolladores a corregir errores.

En la Figura E.27 se puede apreciar los componentes de OwlEye. A grandes rasgos, el procedimiento de detección de problemas se basa en que, dada una captura de pantalla de la IU, que se quiera evaluar, el modelo basado en CNN puede clasificar primero si se relaciona con cualquier problema de visualización a través de la comprensión visual. Una vez que se confirma el problema, el modelo puede localizar la posición detallada del problema en la captura de pantalla de la IU, con el fin de

guiar la corrección del error de usabilidad detectado. Destacar que algunos de los problemas visuales que se consideran son oclusión de componentes, superposición de texto, imagen faltante, valores nulos y pantalla borrosa, entre otros.

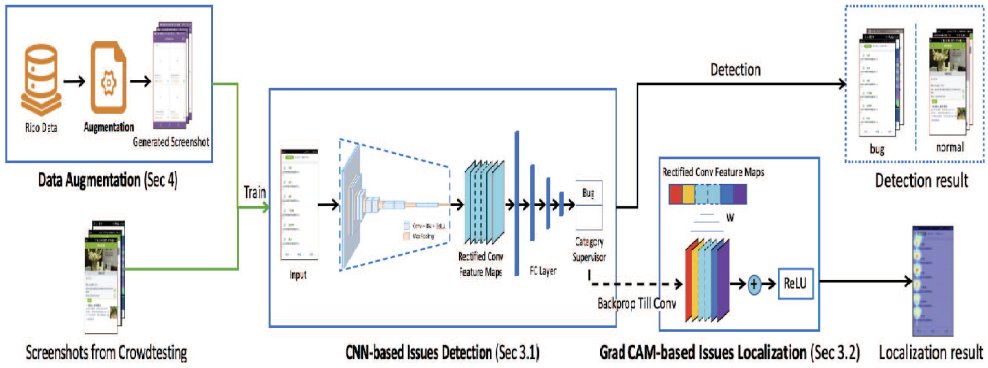


Figura E.27: Vista general de la herramienta OwlEye [Liu *et al.*, 2020].

Para utilizar el modelo de CNN, en necesario entrenarlo. Para esto, se realiza *Data Augmentation* basado en heurística. Liu *et al.* [2020] explica que se desarrolló este método para generar capturas de pantalla de interfaces de usuario con problemas de visualización a partir de imágenes de interfaces de usuario sin errores. Este *data augmentation* se basa en el conjunto de datos de *Rico*, que contiene más de 66.000 capturas de pantalla únicas de 9.300 aplicaciones de Android, así como su archivo JSON adjunto. En la Figura E.28, donde se puede observar que con base en este conjunto de capturas de pantalla, el *data augmentation* propuesto busca generar problemas de visualización tomando las capturas de pantalla de *Rico* para posteriormente entrenar el modelo de CNN. Este proceso se puede apreciar en la Figura E.28.

La Figura E.29 presenta ejemplos de localización de problemas que resaltan las áreas con errores. Se aprecia que OwlEye genera la captura de pantalla con colores a modo de mapa de calor. Las zonas que mayor calor presentan son en las que se detectan potenciales problemas de usabilidad.

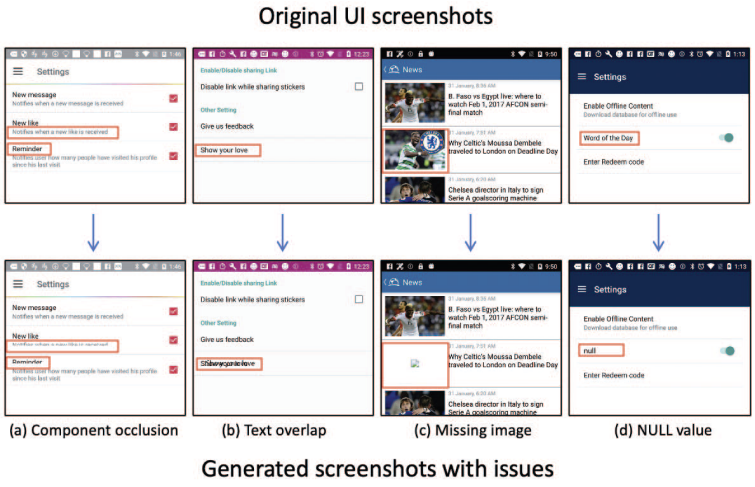


Figura E.28: Ejemplo de *Data Augmentation* Basado en Heurística [Liu *et al.*, 2020].

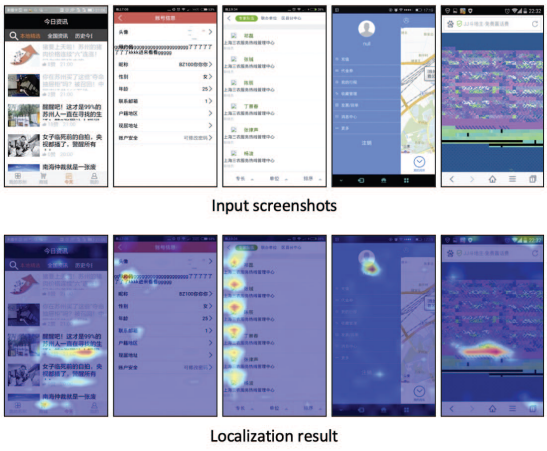


Figura E.29: Ejemplo de localización de problemas de la IU utilizando OwlEye [Liu *et al.*, 2020].

E.12. ADUE (*Automatic Domain Usability Evaluation*)

ADUE es una herramienta que funciona utilizando Java y brinda un enfoque a la usabilidad de dominio, que se basa en el diccionario de dominio y la descripción de las relaciones y propiedades del dominio a través de sus interfaces de usuario [Bačíková *et al.*, 2021]. ADUE detecta problemas de usabilidad de dominio y brin-

da recomendaciones para corregirlos. Esta herramienta pertenece a la categoría de Herramientas que detectan problemas de usabilidad.

Para entender el funcionamiento de ADUE, es necesario explicar la usabilidad de dominio. Según lo expuesto por Bačíková *et al.* [2021], es un enfoque de usabilidad que se centra en cinco aspectos de la IU: el contenido del dominio (términos, relaciones y procesos de la interfaz deben coincidir con los del dominio en el que se diseña la interfaz), la consistencia (las palabras utilizadas en toda la interfaz no deben diferir), idioma utilizado en la interfaz (debe ser el idioma del usuario), especificidad del dominio (la interfaz no debe contener términos demasiado generales) y barreras de lenguaje y errores (la interfaz no debe generar barreras idiomáticas y no debe contener errores lingüísticos). Ante esto, se crean métricas de evaluación de dominio que ayudan a detectar los aspectos mencionados anteriormente en sistemas software. En general, las aplicaciones con una usabilidad de dominio baja tienden a tener una gran cantidad de errores gramaticales, además de términos definidos incorrectamente y falta de información en los lugares donde es necesaria.

Para la extracción de la información del dominio de las interfaces de usuario, se utiliza una herramienta llamada DEAL por su siglas en inglés), la cual realiza este proceso de forma automática. DEAL funciona en dos fases: una de extracción y otra de simplificación. El resultado de la fase de extracción es un modelo de dominio en forma de gráfico que contiene la información del componente de la interfaz que representa el termino extraído, la etiqueta correspondiente al componente, la etiqueta que muestra el componente, su ícono, entre otros que detalla Bačíková *et al.* [2021]. En la fase de simplificación, se filtran los componentes estructurales sin información de dominio.

La herramienta ADUE utiliza DEAL y la información explicada de usabilidad de dominio para funcionar. Para el análisis automatizado de la usabilidad de dominio, Bačíková *et al.* [2021] explican los enfoques que ADUE cubre. Estos enfoques son el análisis ontológico, la evaluación de la especificidad, la evaluación gramatical y el análisis de *tooltip*.

Para el **análisis ontológico**, DEAL extrae información de dominio de la aplicación a la que no le fue realizada una evaluación de la usabilidad de dominio y se exporta a un formato ontológico. Luego, se ejecuta de nuevo DEAL con la nueva versión de la aplicación y se ejecuta el algoritmo de comparación y evaluación de ADUE. Este proceso se puede apreciar en la Figura E.30. Los resultados de la eva-

luación de la ontología se muestran al usuario. ADUE compara la ontología original con la nueva, buscando elementos nuevos, eliminados, modificados y retenidos. La evaluación se realiza considerando el impacto de los cambios realizados en la aplicación. Todos los resultados se muestran al evaluador a modo de lista en el que se pueden ver todos los términos de la aplicación. El evaluador puede seleccionar un término de esta lista para ver los detalles sobre los cambios entre las versiones de ontología junto con los errores o *warnings* correspondientes en caso que se detecte un problema potencial.

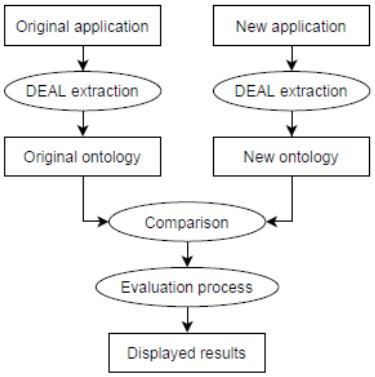


Figura E.30: Representación del proceso de evaluación ontológica usando la herramienta ADUE [Bačíková *et al.*, 2021].

El objetivo de la **evaluación de especificidad** es verificar lingüísticamente las relaciones jerárquicas encontradas en la IU, utilizando diccionarios ontológicos y búsqueda web como fuente de relaciones lingüísticas (utiliza especificamente *WordNet*, *Urban Dictionary* y *Google Web Search*). ADUE recorre todos los elementos del gráfico de modelo de dominio (el cuál es generado por DEAL una vez ejecutado para una aplicación determinada). Para cada grupo de elementos, selecciona los nombres de los términos secundarios y crea un conjunto de palabras secundarias. De cada conjunto de palabras secundarias, elimina todas las formas de pronombres reflexivos y verbos auxiliares para obtener resultados más precisos. ADUE también utiliza el procesamiento del lenguaje natural para reconocer la clase de cada palabra y mantiene solo sustantivos, verbos y adjetivos.

En el enfoque de **evaluación gramatical**, Bačíková *et al.* [2021] explican que hay dos problemas gramaticales comunes que ocurren en interfaces de usuario: una palabra escrita incorrectamente o una palabra que no se tradujo a los idiomas de la IU. Es por esto que la revisión ortográfica se complementa con la revisión de

traducción. Si alguna palabra no se encuentra en el diccionario para el idioma actual, ADUE verifica el idioma predeterminado. Si se encuentra, sus traducciones se agregan a los reemplazos recomendados. De lo contrario, las recomendaciones se basan en palabras similares. Al final del proceso, se proporciona una lista de las correcciones recomendadas al evaluador.

Para el enfoque de **análisis de *tooltip***, ADUE selecciona todos los términos funcionales. Luego, para cada uno, se verifica la presencia de *tooltips*, ya sea inspeccionando el componente que lo representa o verificando la propiedad de descripción del término. Si no se encuentra información de *tooltip*, esta información se agrega a la lista de *warnings* se le recomienda al desarrollador que la agregue. Si no se encuentra información de *tooltip* para algún componente funcional, el resultado se muestra al evaluador de una de dos maneras: como recomendación para agregar información de *tooltip* o como problema de usabilidad de dominio (se determina cuando un componente solo tiene un ícono, o es específico de un solo dominio con solo un ícono o solo tiene una etiqueta de texto). En la Figura E.31 se puede apreciar como la herramienta ADUE destaca los problemas y los muestra al evaluador. En este caso, se señala con una etiqueta roja indicando el componente que se ve afectado por el problema de usabilidad de dominio.

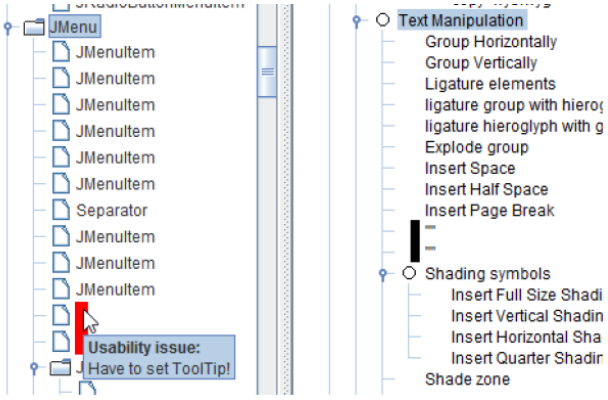


Figura E.31: Ejemplo de elementos de menú de *JSesh* sin información de *tooltip* ni etiqueta. ADUE indica el este problema y entrega una recomendación [Bačíková *et al.*, 2021].

En la Figura E.32 se puede apreciar la interfaz de ADUE. La evaluación ontológica, la evaluación gramatical y la evaluación de especificidad se implementan en esta herramienta. Al realizar el proceso de la evaluación ontológica, los componentes detectados aparecen mostrándose como resultados ante el usuario. ADUE desta-

ca los elementos que se desprenden después de aplicar el proceso de comparación ontológica.

ADUE muestra los diferentes errores detectados usando colores. El rojo se usa para errores gramaticales. El naranja destaca un término principal definido incorrectamente. El color rosa se utiliza para componentes cambiados ilógicamente. El evaluador también puede ver todos los términos que se mantuvieron, agregaron, eliminaron o cambiaron. En todos los casos, ADUE muestra recomendaciones de cambio en la tabla de sugerencias. Se utilizan las métricas de usabilidad de dominio para calcular el puntaje general de usabilidad de dominio de la IU evaluada. Se muestran los errores mostrando la cantidad de estos.

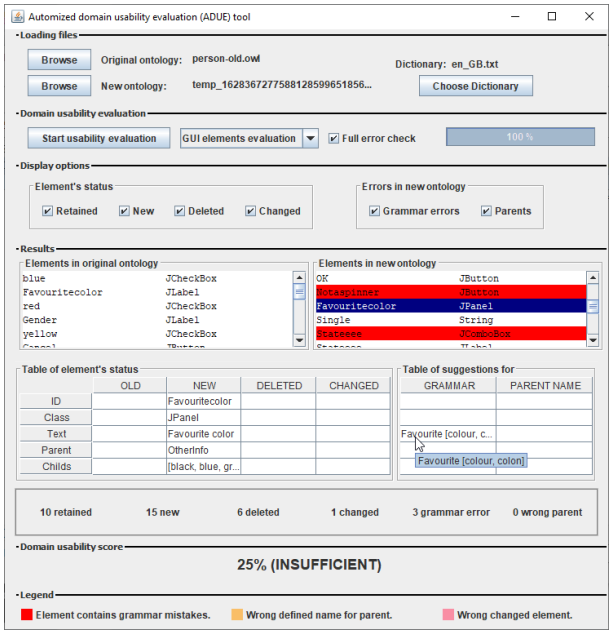


Figura E.32: Ejemplo de elementos de menú de *JSesh* sin información de *tooltip* ni etiqueta. ADUE indica el este problema y entrega una recomendación [Bačíková *et al.*, 2021].

E.13. GTmetrix

GTmetrix es una herramienta online que permite la detección de problemas de usabilidad en entornos web de escritorio. Está basada en *Google's PageSpeed*, *Yahoo's YSlow* y en índices de rendimiento específicos. El concepto de usabilidad que abarca la herramienta es el rendimiento y velocidad de las páginas web analizadas,

por lo que entrega recomendaciones para mejorar en estos aspectos [Al-Sakran y Alsudairi, 2021]. Esta herramienta pertenece a la categoría de Herramientas que detectan problemas de usabilidad.

GTmetrix proporciona informes de análisis de rendimiento de la página web evaluada, entregando además la velocidad de carga de la página, sugerencias de mejora y puntajes de rendimiento general. Gracias a las funcionalidades entregadas por *YSlow*, la herramienta permite rastrear un sitio web y compararlo con una lista de 23 reglas basadas en las reglas de Yahoo para sitios web de alto rendimiento. Luego, califica el sitio web según estas 23 reglas y otorga a los usuarios un puntaje general basándose en el promedio. La escala de clasificación es de 0 a 100. La herramienta muestra esta puntuación, detallando también el porcentaje de cumplimiento de cada regla y los compara con los promedios de los valores establecidos por las reglas de Yahoo. Otra opción que permite la herramienta es calcular el tiempo total necesario para cargar el contenido completo de una página, sumándose a los indicadores de rendimiento que ocupa GTmetrix para determinar el rendimiento de la página web analizada.

Al detectar problemas durante el análisis de un sitio web, GTmetrix entregará recomendaciones para mejorar y corregir dichos problemas. Algunas recomendaciones que puede ofrecer la herramienta son la optimización y escala de imágenes, aprovechar el almacenamiento en caché del navegador, evitar redireccionamientos a la página de destino, entre otros. Ante esto, GTmetrix relaciona las recomendaciones con los problemas de usabilidad detectados, mostrándolos como *warnings*. Estos problemas son mostrados al evaluador para que pueda realizar las correcciones pertinentes.

En la Figura E.33 se puede apreciar la interfaz de GTmetrix. En esta interfaz, la herramienta muestra el sitio web que está siendo analizado, junto con el puntaje que obtuvo con base en su rendimiento general. GTmetrix también muestra de forma visual la velocidad de carga del sitio web, mostrando los tiempos en los que se cargan los elementos de la IU del sitio web y en qué momento está completamente cargado.

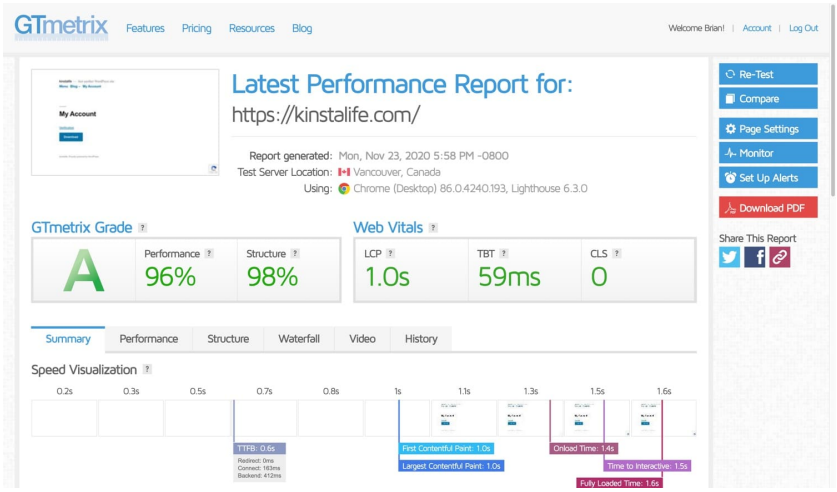


Figura E.33: Interfaz usuario de GTmetrix[1].

E.14. Dareboost

Dareboost es una herramienta que permite la detección de problemas de usabilidad en entornos web de dispositivos móviles. Dareboost también permite evaluar la accesibilidad de sitios web. Esta herramienta mide las métricas de rendimiento relacionadas con los elementos que conforman la página web a evaluar, detectando problemas y entregando información de los análisis realizados a los evaluadores sobre los sitios web evaluados [Al-Sakran y Alsudairi, 2021]. Esta herramienta pertenece a la categoría de Herramientas que detectan problemas de usabilidad.

Dareboost mide las métricas de rendimiento de las debilidades, *warnings*, éxitos, tiempos de carga, tamaño total de la página web evaluada y la cantidad de solicitudes HTTP, presentando resultados para cada factor detectado. Dareboost proporciona informes de resultados de pruebas con un puntaje general de página, número de problemas, mejoras recomendadas y éxito de la usabilidad del sitio web analizado. Además, Dareboost ofrece opciones de pruebas de velocidad del sitio web móvil que se desea evaluar y brinda opciones de geolocalización global que afectan a los resultados de la prueba, el tipo de navegador utilizado y los sistemas operativos para distintos móviles. La herramienta muestra puntajes de los sitios web en función de sus métricas y brinda sugerencias sobre cómo se puede mejorar la usabilidad del sitio web analizado.

[1]https://kinsta.com/es/blog/gtmetrix-herramienta-de-test-de-velocidad/

En la Figura E.34 se puede apreciar la interfaz de Dareboost. La herramienta muestra un puntaje general del sitio web analizado, mostrándolo con un porcentaje. Junto con esto, muestra la cantidad de errores, recomendaciones de mejoras y éxitos. Se muestra que navegador se está utilizando, la simulación de geolocalización correspondiente y la velocidad de carga del sitio web analizado. Por último, Dareboost entrega información acerca de la cantidad de solicitudes realizadas, el peso en kilobytes y tiempos importantes de carga del sitio web: el primer byte cargado, el tiempo en el que se empieza a renderizar el sitio web y cuando está completamente cargado.

Figura E.34: Interfaz de usuario de Dareboost[2].

[2]https://www.dareboost.com/en