



UNIVERSIDAD
NACIONAL
DE LA PLATA

FACULTAD DE INFORMÁTICA

TESINA DE LICENCIATURA

TÍTULO: Evaluación del esfuerzo del usuario en la web móvil

AUTORES: Lautaro Ezequiel Aria – Fermin Moreno

DIRECTOR/A: Julián Grigera

CODIRECTOR/A: Alejandra Garrido

ASESOR PROFESIONAL: Juan Cruz Gardey

CARRERA: Licenciatura en Informática

Resumen

La popularidad de los dispositivos móviles ha transformado la forma en que interactuamos con la web, convirtiéndose en el principal punto de acceso a Internet para una gran parte de la población mundial. En esta tesis, proponemos una aproximación para evaluar y mejorar la usabilidad de las interfaces web móviles, centrándonos en el esfuerzo de interacción del usuario al completar formularios.

Palabras Clave

Usabilidad, interacción del usuario, dispositivos móviles, evaluación del esfuerzo, interfaces web.

Conclusiones

Con este trabajo se sientan las bases para una evaluación más automatizada y precisa de la usabilidad en dispositivos móviles, con el potencial de mejorar significativamente la experiencia del usuario en este contexto.

Trabajos Realizados

Se desarrollaron y adaptaron métricas para evaluar el esfuerzo de interacción en dispositivos móviles. Se realizaron pruebas con 14 voluntarios, y se recopilieron datos que fueron analizados para refinar las métricas.

Trabajos Futuros

Sugerimos usar nuestros resultados para entrenar un modelo de predicción que sirva para obtener de manera automática el esfuerzo requerido de cualquier formulario web en dispositivos móviles.

Fecha de la presentación:

1. Introducción

1.1 Esfuerzo de interacción

1.2 Resultados Obtenidos

1.3 Estructura del Informe

2. Trabajo Relacionado

2.1 Estado del Arte

2.2 Contexto

3. Desarrollo de Métricas para Dispositivos Móviles

3.1 Adaptación de métricas originales

3.2 Evaluación con Participantes

3.2.1 Preparación

Detalles específicos sobre la implementación

Diferencias con la versión desktop

Backend

Infraestructura para presentar las aplicaciones de prueba

3.2.2 Ejecución

3.2.3 Resultados

3.2.4 Amenazas a la validez

3.2.5 Conclusiones

4. Refinamiento del Listado de Métricas

4.1 Preparación

4.2 Ejecución

4.3 Listado Refinado

4.4 Amenazas a la validez

4.5 Métricas agregadas luego del análisis de los expertos

4.6 Conclusiones

5. Conclusiones

5.1 Diferencias y Similitudes entre las Métricas Móviles y de Escritorio

5.2 Medición del Esfuerzo en Dispositivos Móviles

5.3 Percepción del Esfuerzo en Dispositivos Móviles

6. Contribuciones y Futuras Direcciones

Anexo A: Detalles Técnicos y Herramientas Utilizadas

Referencias

Capítulo 1

Introducción

La popularidad de los dispositivos móviles ha transformado la forma en que interactuamos con la web, convirtiéndose en el principal punto de acceso a Internet para una gran parte de la población mundial. Las aplicaciones móviles tienen un alcance más amplio que las aplicaciones web de escritorio; en un estudio realizado por la (UIT) en 2017 se mostró que con una población mundial de 7.400 millones de personas, en el mundo había 7.700 millones de suscripciones a teléfonos móviles¹, es decir que hallaron más dispositivos de este tipo que habitantes. En contraste con esto, el número de computadoras de escritorio es mucho menor.

Es crucial destacar que los dispositivos móviles tienen además una adopción mucho mayor que los equipos de escritorio y son utilizados con mayor frecuencia por los usuarios. En un estudio realizado en 2022² se halló que el 55% de las visitas online a nivel mundial proceden de dispositivos móviles, el 43% de computadoras y el 2% de tabletas. Esto resalta la necesidad de investigar y entender la interacción de los usuarios de dispositivos móviles, ya que esta experiencia puede diferir significativamente de las aplicaciones de escritorio. Para avanzar en este ámbito y mejorar la calidad de las evaluaciones de usabilidad en dispositivos móviles, es esencial realizar estudios específicos para éstos, lo que motiva la propuesta de esta tesis.

Esta tendencia de la población a utilizar dispositivos móviles ya ha impulsado la necesidad de desarrollar y optimizar la usabilidad de las interfaces web móviles, con el objetivo de garantizar una experiencia de usuario satisfactoria y competitiva. A pesar de los beneficios evidentes de abordar la usabilidad web, todavía existe una brecha en la asignación de recursos hacia esta práctica en comparación con otras áreas del desarrollo de software Hinderks et al (2019)

Para abordar esta necesidad, han surgido herramientas automáticas y procesos que buscan reducir los costos asociados con la evaluación y reparación de la usabilidad Oulasvirta et al (2018); Nebeling et al (2013).

En el campo de la evaluación de la usabilidad y la experiencia del usuario en dispositivos móviles, han surgido diversas herramientas que buscan mejorar la calidad de las interfaces y la satisfacción del usuario. Entre ellas, se encuentran trabajos como los mencionados anteriormente Speicher et al (2014), cada una con enfoques y funcionalidades específicas que abordan diferentes aspectos de la experiencia del usuario en dispositivos móviles.

¹ “En el mundo hay más celulares que humanos.”
<https://otitelecom.org/telecomunicaciones/mundo-mas-celulares-humanos/>

² ¿Qué Porcentaje del Tráfico de Internet es Móvil?
<https://kinsta.com/es/movil-vs-escritorio-cuota-mercado/#:~:text=Si%20nos%20fijamos%20en%20una,y%20parece%20que%20ten%C3%ADan%20raz%C3%B3n.>

Oulasvirta et al. (2018) proponen AIM, que ofrece un servicio en línea que combina varios modelos y métricas validados empíricamente sobre percepción y atención del usuario, facilitando la evaluación de diseños de interfaces gráficas (GUI) con desgloses detallados, visualizaciones y comparaciones estadísticas. Si bien AIM proporciona una amplia gama de métricas de usabilidad, se centra principalmente en aspectos generales de la experiencia del usuario y la percepción visual.

Nebeling et al. (2013) presentan W3Touch como una herramienta que ayudaría a mejorar las páginas web con respecto a pantallas táctiles, por eso consideran métricas como el zoom y el *misclicks*. De esta manera proponen mejorar los tamaños y posicionamientos de los distintos elementos en la página.

En el estudio realizado por Speicher et al. (2014), se emplea un enfoque similar al nuestro al utilizar la herramienta WaPPU para medir la usabilidad y predecir la calidad de la experiencia del usuario en páginas web.

Mientras que la mayoría de estas herramientas pueden identificar problemas generales de usabilidad, pocas se enfocan en problemas de interacción más específicos y localizados, que pueden ser más manejables, económicos y susceptibles de automatización. Esto plantea la posibilidad de combinar enfoques ágiles y centrados en el usuario, permitiendo una mejora incremental y continua en la usabilidad de las interfaces web móviles Badran et al. (2018).

Como en varios métodos actuales que combinan el análisis de arquitectura de software y la evaluación de usabilidad Bel et al. (2010), Bass et al. (2003) nuestra propuesta busca aprovechar el potencial de combinar estos enfoques para obtener resultados más completos y significativos en una etapa temprana del proceso de desarrollo de software iterativo. Al considerar tanto la arquitectura de software como el esfuerzo del usuario al completar formularios, podemos mejorar la calidad y la cantidad de los resultados de la evaluación, lo que potencialmente reduce la necesidad de múltiples iteraciones en el desarrollo del software. En última instancia, esta combinación puede elevar el estándar de las metodologías de evaluación existentes y contribuir al avance del estado actual de las técnicas de evaluación en el ámbito de la usabilidad de dispositivos móviles.

En esta tesis, proponemos **una aproximación para evaluar y mejorar la usabilidad de las interfaces web móviles, centrándonos en el esfuerzo de interacción del usuario** al completar formularios. Nuestra propuesta incluye el desarrollo de una plataforma web integral con formularios diseñados específicamente para dispositivos móviles, así como la creación y aplicación de métricas para medir el esfuerzo del usuario en esta tarea.

En nuestra búsqueda por desarrollar metodologías más precisas para la evaluación de la usabilidad en dispositivos móviles, hemos encontrado inspiración y evidencia en estudios previos que han abordado la medición de la experiencia del usuario de manera integral. Un ejemplo significativo es el Standardized User Experience Percentile Rank Questionnaire (SUPR-Q) desarrollado por Sauro et al. (2015), un instrumento consolidado que ha sido validado a través de un estudio exhaustivo de cinco años y respuestas de 4,000 usuarios sobre

más de 100 sitios web. El SUPR-Q, que mide factores críticos como la usabilidad, confianza, apariencia y lealtad, ha demostrado tener una alta consistencia interna y validez convergente con cuestionarios existentes, como el System Usability Scale (SUS).

La metodología aplicada en el SUPR-Q, especialmente su enfoque en generar puntuaciones normalizadas que permiten comparaciones benchmarking eficaces, proporciona un modelo valioso para nuestra investigación. Este enfoque nos ha impulsado a considerar no solo la adaptabilidad de las métricas de usabilidad existentes a contextos móviles, sino también la importancia de escuchar las respuestas de usuarios para entender cómo las interfaces móviles se comparan con otras en términos de esfuerzo de interacción. De esta manera, orientándonos hacia la creación de una plataforma que no solo evalúa la usabilidad, sino que también contextualiza los resultados dentro de un marco comparativo más amplio para los investigadores y profesionales del campo.

Para medir el esfuerzo de interacción trabajamos sobre el concepto y las métricas creadas por Grigera et al. (2019). En este artículo los autores proponen una métrica unificada para evaluar *widgets* de interfaz individuales por su esfuerzo de interacción, es decir, el nivel de esfuerzo requerido por el usuario para interactuar con un widget único, según lo percibe un experto. Esto permite incluso comparar widgets diferentes que tienen el mismo propósito, como por ejemplo la selección de opciones mediante listas vs. *radio buttons*. Las métricas se basan en medidas (microcomportamientos) que se pueden capturar automáticamente a partir de la interacción del usuario. Su objetivo es evaluar el esfuerzo de forma automática, y para eso entrenaron un modelo de predicción en base a las métricas, clasificadas con puntajes asignados por expertos que juzgaron el esfuerzo para cada widget completado por los usuarios.

En nuestro trabajo adaptamos las métricas originalmente creadas para escritorio a interfaces de dispositivos móviles y, además, creamos algunas específicas para este contexto. De la misma manera que en el trabajo original, también contamos con la ayuda de expertos en UX que se encargaron de puntuar el esfuerzo requerido de los voluntarios que completaron los formularios de prueba. Por último, la etapa de entrenar un modelo de predicción queda pendiente para futuros trabajos. Para esto creamos un script que exporta la información de modo de dejarla lista en un formato que permita este entrenamiento.

En conclusión, con este trabajo se sientan las bases para una evaluación más automatizada y precisa de la usabilidad en dispositivos móviles, con el potencial de mejorar significativamente la experiencia del usuario en este contexto.

1.1 Esfuerzo de interacción

Nuestro trabajo está basado en la noción de “esfuerzo de interacción”. Este concepto se asocia a la medida del trabajo o dificultad percibida por un usuario al interactuar con los distintos elementos de un formulario. En los estudios realizados por [Grigera et al. \(2019\)](#) y [Gardey et al. \(2022\)](#), se propone una metodología para medir el esfuerzo de interacción en formularios web. Este esfuerzo de interacción fue evaluado mediante la observación de cómo los usuarios

interactuaban con diferentes elementos de un formulario, asignándoles un puntaje del 1 al 4 según la dificultad percibida. Un puntaje bajo indica una interacción sencilla, mientras que un puntaje alto refleja una interacción más compleja que demanda más esfuerzo. En estos trabajos, se entrenaron modelos de predicción a partir del puntaje asignado manualmente por expertos sobre capturas de pantalla, para luego lograr predecir dicho puntaje en base a capturas de eventos de interacción.

La predicción del puntaje de esfuerzo está basada en métricas detalladas que describen cómo interactúa el usuario con el widget, lo que también permite entender mejor qué acciones llevaron a dicho puntaje. Estas métricas incluyen aspectos como la velocidad de cursor, la precisión del toque, la cantidad de correcciones realizadas en campos de texto y otros indicadores relevantes para la evaluación del esfuerzo de interacción.

El objetivo de nuestro enfoque consiste en utilizar **métricas desarrolladas específicamente para dispositivos móviles**, con el objetivo final de entrenar un modelo de predicción capaz de calcular el esfuerzo de interacción en estos dispositivos. Para hacer este modelo harían falta pruebas con una mayor cantidad de usuarios y luego permitiría estimar el esfuerzo de interacción sin necesidad de la intervención directa de expertos en UX.

Para el cálculo del esfuerzo de interacción, consideramos todas las métricas relevantes para evaluar el esfuerzo del usuario al completar formularios en dispositivos móviles. Nos basamos en la premisa de que cada acción realizada por el usuario en un widget contribuye de manera única al esfuerzo total, por lo que es crucial que todas las métricas pertinentes fueron tomadas en cuenta. En este sentido, nuestra metodología se inspira particularmente en el trabajo de [Gardey et al. \(2022\)](#), sin embargo, dada la naturaleza de la interacción en dispositivos móviles, fue necesario adaptar y crear nuevas métricas específicas para este contexto.

Durante el proceso de desarrollo de las métricas, llevamos a cabo un análisis exhaustivo que incluyó la revisión de métricas existentes para dispositivos de escritorio, la modificación de algunas de ellas y la creación de nuevas métricas, todo ello en función de las necesidades y particularidades de la interacción en dispositivos móviles. Este proceso fue complementado con la revisión y validación por parte de expertos en Experiencia de Usuario (UX), quienes aportaron sus conocimientos y experiencia para refinar y mejorar nuestras métricas, asegurando así su relevancia y eficacia en la evaluación del esfuerzo de interacción.

Como resultado, hemos logrado adaptar y ampliar el enfoque basado en la web de escritorio para abordar de manera más completa los desafíos específicos de la interacción en dispositivos móviles. Esta adaptación incluye la consideración de aspectos como el tamaño de pantalla, la precisión táctil y la variabilidad en el contexto de uso, que son distintivos de la experiencia del usuario en dispositivos móviles.

1.2 Resultados Obtenidos

A partir del estudio de métricas para interfaces web para escritorio, adaptamos y desarrollamos métricas específicas para dispositivos móviles. Estas métricas fueron refinadas y validadas en colaboración con expertos en Experiencia de Usuario (UX), asegurando su relevancia y precisión en la medición del esfuerzo de interacción. Este proceso nos permitió ajustar las métricas de usabilidad de manera efectiva y proporcionar evaluaciones precisas.

Para lograr esto realizamos un estudio en el cual reclutamos a 14 voluntarios, cuyas interacciones fueron grabadas en video mientras completaban cinco formularios diferentes, proporcionando una buena fuente de datos. Estos videos fueron utilizados para que los expertos hagan una evaluación profunda de la experiencia del usuario.

Las evaluaciones de los expertos nos revelaron la necesidad de implementar nuevas métricas, diseñadas para capturar comportamientos específicos y errores comunes durante la interacción del usuario. Estas nuevas métricas enriquecieron nuestro entendimiento de la interacción del usuario y fortalecieron nuestra metodología, dando lugar finalmente a un conjunto de métricas refinado.

Finalmente, implementamos algunas métricas nuevas que descubrimos en el proceso pero no estaban ya implementadas en su versión de escritorio para que pudiéramos realizar una adaptación.

Este enfoque integrado no sólo proporcionó una comprensión detallada del esfuerzo del usuario, sino que también estableció una base sólida para futuras investigaciones y desarrollos en la usabilidad de interfaces móviles, mejorando significativamente la experiencia del usuario en entornos digitales.

1.3 Estructura del Informe

El trabajo comienza con la búsqueda de trabajos relacionados con la experiencia de usuario en dispositivos móviles. Encontramos muchos trabajos que abordan el tema de distintas maneras, enfocados en distintas cosas y no en la medición del esfuerzo, como lo es en nuestro caso ([2 Trabajo relacionado](#)).

Para llevar a cabo la tarea que se planteó, se comenzó con la adaptación de los formularios heredados del trabajo de [Gardey et al. \(2022\)](#) para dispositivos móviles ([3.2.1 Detalles específicos sobre la implementación](#)), esto incluye en total una cantidad de cinco formularios. Luego detectamos que métricas heredadas del trabajo de Gardey nos iban a servir y cuáles no, las útiles tuvimos que adaptarlas a dispositivos móviles ([3.1 Adaptación de métricas a dispositivos móviles](#)) y luego pensar que nuevas métricas agregar en este contexto. Las pruebas para determinar esto fueron realizadas por nosotros, requirió de experimentar muchas veces con los formularios para tomar las mejores decisiones. Una vez que quedamos conformes con las métricas a calcular procedimos a desarrollarlas e hicimos una comparación con las métricas heredadas ([3.2.1 Diferencias con la versión desktop](#)).

Posteriormente necesitamos almacenar la información de estas métricas, por lo que decidimos utilizar una forma de base de datos simple como lo es SQLite. Luego de almacenar esta información desarrollamos un script en Python para poder exportarla en un archivo CSV, con el objetivo de que el mismo pueda ser utilizado para hacer un modelo de predicción en un futuro si se quisiera ([3.2.1 Backend](#)).

Con el flujo de trabajo completo, es decir, recopilación de métricas y almacenamiento persistente, ya estábamos en condiciones de realizar pruebas con usuarios reales, para obtener datos realmente relevantes y no sesgados por nosotros. Entonces reclutamos como voluntarios a familiares y amigos que estaban dispuestos a hacer las pruebas completando los 5 formularios realizados. Este procedimiento consistió en grabarlos completando las pruebas, de esta manera obtuvimos videos donde se analizaría que se les dificultó a la hora de rellenar los formularios ([3.2.2 Ejecución](#)).

Con un total de 14 videos nos contactamos con dos expertos en UX para que nos puedan dar sus apreciaciones basándose en estos videos. Coordinamos una reunión virtual con cada uno de ellos para que nos puedan explicar que percibieron en los videos, en que se basaron para determinar si a un usuario le había costado completar cualquier widget ([4 Refinamiento del Listado de Métricas](#)).

Con sus devoluciones encontramos valioso agregar dos métricas nuevas al listado de métricas original, analizamos de qué manera desarrollarlo y pasamos a la acción, todo con el fin de mejorar los resultados finales ([4.3 Listado refinado](#)).

En conclusión la versión en dispositivos móviles presentó algunas métricas distintas a la versión de desktop, ya que la manera de operar en móviles es diferente, al mismo tiempo algunas otras métricas de desktop son similares ([5.1 Diferencias y similitudes](#)).

Estas métricas desarrolladas específicamente para móviles proporcionaron una manera eficaz de calcular la medición del esfuerzo y comprender mejor la experiencia del usuario ([5.2 Medición del esfuerzo en dispositivos móviles](#)).

Es decir que para mejorar la experiencia de usuario la percepción del esfuerzo debe ser abordada de manera distinta, tanto en desktop como en móviles, para optimizar la experiencia de usuario ([5.3 Percepción del esfuerzo en dispositivos móviles](#)).

Como paso final en nuestra tesis sugerimos usar nuestros resultados para entrenar un modelo de predicción, que sirva para obtener de manera automática el esfuerzo requerido de cualquier formulario web en dispositivos móviles ([6. Contribuciones y futuras direcciones](#)).

Capítulo 2

Trabajo Relacionado

En el ámbito de la experiencia del usuario (UX) en dispositivos móviles, la evaluación de la usabilidad se ha convertido en un tema central de investigación. La creciente prevalencia de aplicaciones móviles y sitios web adaptados a dispositivos móviles ha generado la necesidad de comprender y mejorar la interacción del usuario con estos dispositivos. En este contexto, la revisión de trabajos previos se vuelve crucial para identificar métricas clave, metodologías efectivas y hallazgos relevantes que puedan informar y enriquecer futuras investigaciones.

Una parte fundamental de esta revisión es el análisis de estudios que abordan la evaluación de la usabilidad en dispositivos móviles. Se han identificado proyectos que exploran la evaluación de la usabilidad de aplicaciones móviles desde diversas perspectivas, como la medición del esfuerzo del usuario, la estimación del esfuerzo de desarrollo y prueba de aplicaciones móviles, y la evaluación automatizada de la usabilidad.

Entre estos proyectos se encuentra el trabajo llevado a cabo por Hussain et al. (2012), que se centra en el desarrollo de un modelo basado en métricas para evaluar aplicaciones móviles. Este estudio destaca la necesidad de establecer métricas sólidas y efectivas para medir la usabilidad en dispositivos móviles. Sin embargo, la falta de enfoque específico en la interacción del usuario con formularios en dispositivos móviles ha motivado la realización de investigaciones adicionales en este campo.

En este capítulo mostraremos, por un lado, los trabajos relevantes en el área ([Sección 2.1](#)). Por otra parte, describiremos trabajos importantes para dar contexto a nuestra propuesta en particular.

2.1 Estado del Arte

En esta sección nos enfocamos en explorar estudios anteriores que se centran en la evaluación de la experiencia del usuario en formularios en dispositivos móviles, con el objetivo de identificar métricas clave, metodologías y hallazgos relevantes que puedan informar y enriquecer nuestro enfoque.

La revisión de trabajos relacionados no solo nos permitió comprender los avances previos en el campo de la usabilidad de aplicaciones móviles, sino que también nos ayudó a identificar áreas que la investigación actual no ha explorado, evitando la duplicación de esfuerzos y proporcionando una base sólida sobre la cual construir nuestra investigación

En el marco de nuestro proyecto, encontramos un trabajo relevante llevado a cabo por Hussain et al. en (2012), que se centró en la evaluación de la usabilidad de aplicaciones móviles, un tema que también es central en nuestra investigación. Al igual que nosotros, Hussain y su equipo identificaron un desafío importante: la falta de directrices claras y métricas específicas en el ámbito de los dispositivos móviles. Su enfoque se centró en revisar las métricas de usabilidad existentes y desarrollar un modelo basado en métricas para evaluar aplicaciones móviles.

La similitud entre su trabajo y el nuestro radica en la necesidad de establecer métricas sólidas y efectivas para medir la usabilidad de las aplicaciones móviles. Sin embargo, en nuestro caso, hemos dado un paso adicional al desarrollar una plataforma web diseñada específicamente para evaluar la experiencia del usuario en formularios en dispositivos móviles.

Janlert et al. (2017) propone un enfoque sistemático y analítico para comprender y medir la interactividad, basado en conceptos claramente definidos y, cuando sea posible, propiedades cuantificables. Mientras que en este artículo la contribución principal son una serie de definiciones y términos, junto con un intento inicial de enmarcar las condiciones de interacción e interactividad, por otro lado, nuestra tesis se enfoca en el desarrollo de métricas específicas para medir el esfuerzo del usuario al completar formularios en dispositivos móviles, lo cual se relaciona indirectamente con la interactividad. Nuestro trabajo se enfoca en un aspecto específico de la interacción en dispositivos móviles con las métricas. Ambos trabajos comparten la preocupación por comprender y medir conceptos clave en la interacción humano-computadora, pero se enfocan en aspectos diferentes y complementarios de este campo de estudio.

Nuestra plataforma ha permitido la recopilación de datos significativos sobre la interacción del usuario con formularios en dispositivos móviles, lo que a su vez ha revelado información valiosa sobre las métricas clave que impactan en la experiencia del usuario. Estos hallazgos son esenciales para mejorar el diseño de la experiencia del usuario en aplicaciones móviles y sitios web adaptados a dispositivos móviles.

A menudo, los enfoques de análisis de interacción de usuario consideran a todos los usuarios de manera uniforme como el trabajo de cálculo del esfuerzo de interacción del widget individualmente hecho por Grigera et al. (2019), lo que puede no reflejar adecuadamente las variaciones en la experiencia y el esfuerzo de diferentes usuarios al interactuar con formularios. Existen casos donde sí se tuvo en cuenta esto como en el estudio de velocidad de interacción de usuario en la web de Bonora et al. (2022) y se obtienen métricas más realistas, este proyecto tratará de apuntar a algo similar pero con dispositivos móviles sobre la web.

En un trabajo de evaluación de técnicas de usabilidad para celulares Kjeldskov et al. (2004) los autores se concentran en desarrollar técnicas de evaluación de usabilidad en un entorno de laboratorio. Esta propuesta aborda la evaluación del esfuerzo del usuario al interactuar con formularios específicos en la web móvil. Ambos proyectos están enmarcados en el ámbito de la usabilidad móvil y buscan proporcionar recomendaciones prácticas para mejorar la interacción del usuario con sistemas móviles.

La investigación que se presenta en este artículo de [Cui et al \(2011\)](#) se centra en la percepción de las actividades interactivas en dispositivos móviles y su influencia en el diseño del historial de navegación. Se destacan tres hallazgos relevantes que tienen implicaciones importantes para nuestro proyecto de tesis relacionado con la evaluación de la experiencia del usuario en dispositivos móviles. En primer lugar, la importancia de los objetos de contenido en la percepción de la interacción móvil se alinea con nuestro enfoque en la evaluación de formularios y widgets en nuestra plataforma. Al igual que los objetos contenidos en el artículo, los widgets en nuestra plataforma son elementos cruciales en la interacción del usuario y la medición de su esfuerzo. En segundo lugar, la idea de utilizar jerarquías y la similitud en contenido para organizar segmentos de interacción se relaciona con cómo organizamos y priorizamos eventos en nuestra plataforma. La agrupación de segmentos y la identificación de eventos representativos son aspectos cruciales para la medición de métricas de esfuerzo. Finalmente, la baja consistencia entre los participantes en la percepción de eventos destaca la importancia de adaptar nuestros diseños y métricas a las diferencias individuales. Nuestra plataforma debe ser lo suficientemente flexible para acomodar las variaciones en la experiencia del usuario y brindar resultados significativos.

El artículo de [Kaur et al \(2019\)](#), se enfoca en la estimación del esfuerzo de prueba en el desarrollo de aplicaciones móviles. Este estudio aborda cómo las características únicas de las aplicaciones móviles influyen en la estimación del esfuerzo de prueba, un aspecto crucial para el desarrollo eficiente de software móvil.

Aunque ambos trabajos se sitúan en el contexto de la tecnología móvil, nuestras orientaciones son distintas. Mientras que nosotros evaluamos la interacción del usuario con la interfaz móvil, el trabajo de [Kaur et al \(2019\)](#) se concentra en los procesos de desarrollo y prueba de aplicaciones móviles. Sin embargo, hay un punto de convergencia interesante: la importancia de considerar las características específicas de los dispositivos móviles. En nuestro caso, estas características influyen en cómo los usuarios interactúan con los formularios, mientras que en el otro estudio, afectan la manera en que se debe estimar el esfuerzo para probar las aplicaciones.

Además, ambos estudios utilizan metodologías similares, como la revisión sistemática de la literatura y las encuestas, para recopilar y analizar datos. Esto refleja una tendencia común en la investigación de tecnologías móviles, donde la combinación de análisis teórico y feedback práctico es fundamental para obtener resultados relevantes y aplicables.

El artículo de [Anderson et al \(2001\)](#) se centra en la personalización de sitios web para usuarios móviles. Este enfoque implica adaptar el contenido y la interfaz de un sitio web para mejorar la experiencia del usuario en dispositivos móviles, teniendo en cuenta las limitaciones y necesidades específicas de estos dispositivos, como pantallas pequeñas y conexiones de red más lentas.

Aunque nuestros enfoques son distintos, ambos estudios subrayan la importancia de optimizar la experiencia web en dispositivos móviles. Mientras que nosotros evaluamos la interacción del usuario con la interfaz móvil, el artículo de la ACM explora cómo la personalización del sitio web puede mejorar la usabilidad y la satisfacción del usuario.

Hay un punto de convergencia interesante en ambos estudios: la necesidad de considerar las características específicas de los dispositivos móviles. En nuestro caso, estas características influyen en cómo los usuarios interactúan con los formularios, mientras que en el estudio de la ACM, afectan la manera en que se debe personalizar el contenido web para cada usuario.

En ambos estudios se usan metodologías que incluyen análisis de interacción del usuario y pruebas de usabilidad, aunque con diferentes aplicaciones. Nuestro estudio se centra en medir y entender el esfuerzo del usuario, mientras que el artículo de la ACM se enfoca en adaptar el contenido web para mejorar la experiencia del usuario.

Por otro lado, el artículo de [Kaur et al \(2022\)](#) aborda la estimación del esfuerzo de desarrollo y prueba de aplicaciones móviles desde una perspectiva de desarrollo de software. Este estudio destaca la importancia de adaptar las técnicas de estimación para abordar las características únicas del software móvil, diferenciándolo del software tradicional. Además, sugiere la necesidad de desarrollar modelos formales específicos para la estimación en aplicaciones móviles, teniendo en cuenta estas características distintivas.

Ambos estudios reconocen la importancia y la singularidad del entorno móvil. Mientras que el artículo se enfoca en el proceso de desarrollo y prueba de software, nosotros nos centramos en la experiencia del usuario final. Sin embargo, ambos trabajos subrayan la necesidad de considerar las peculiaridades de los dispositivos móviles, ya sea en términos de desarrollo de software o en la interacción del usuario con la interfaz.

Además, la investigación del artículo sobre las metodologías ágiles y las técnicas de estimación en el desarrollo de software móvil complementa nuestro enfoque en la evaluación del esfuerzo del usuario. Entender las diferencias en el desarrollo de software para móviles puede proporcionar introspectivas valiosas para nuestro estudio, especialmente en términos de cómo estas diferencias afectan la experiencia del usuario al interactuar con formularios web.

El artículo de [Lettner et al \(2011\)](#) se centra en la evaluación de la usabilidad de aplicaciones móviles, un proceso que puede ser bastante prolongado, especialmente porque los estudios de campo suelen ser más adecuados que las evaluaciones de laboratorio. Este trabajo presenta un enfoque innovador para la evaluación automatizada de la usabilidad de una aplicación móvil, basado en la observación y análisis del comportamiento del usuario al utilizar la aplicación en condiciones reales. El objetivo es ayudar a los desarrolladores a hacer que las aplicaciones móviles sean más fáciles de aprender, de usar y estéticamente más agradables, lo que conduciría a una mayor usabilidad y aceptación de las aplicaciones.

Comparando este enfoque con nuestro proyecto, encontramos algunas similitudes y diferencias clave. Ambos estudios reconocen la importancia de la usabilidad en el entorno móvil y buscan formas de mejorar la experiencia del usuario. Mientras que Lettner y Holzmann se enfocan en un marco para la evaluación automatizada de la usabilidad, nuestro proyecto se centra específicamente en la evaluación del esfuerzo del usuario al interactuar con formularios en sitios web móviles.

Otra similitud es la importancia de la observación del comportamiento del usuario en condiciones reales, un aspecto que ambos estudios consideran crucial para una evaluación precisa de la usabilidad o el esfuerzo del usuario. Sin embargo, nuestro proyecto podría beneficiarse de la integración de algunos de los enfoques automatizados propuestos por Lettner y Holzmann para complementar y enriquecer nuestro análisis del esfuerzo del usuario.

2.2 Contexto

Una métrica de esfuerzo de interacción ya ha sido desarrollada para interfaces de escritorio. El proyecto de [Grigera et al. \(2019\)](#) propone una métrica unificada para comparar el esfuerzo de interacción requerido por los usuarios al interactuar con diferentes widgets dentro de una interfaz. Esta métrica se basa en micro-medidas capturadas automáticamente de los registros de interacción, lo que permite su predicción automática.

Los autores desarrollaron un modelo de puntuación unificado llamado "esfuerzo de interacción" para evaluar el nivel de esfuerzo requerido por los usuarios. Esto se logra mediante el uso de medidas automáticas y modelos de aprendizaje automático, lo que representa un avance significativo en la automatización del proceso de evaluación de la usabilidad. Utilizando árboles de decisión, el estudio muestra los resultados de predecir dicha puntuación, lo que permite comparar cómo diferentes widgets realizan una misma tarea y evaluar su usabilidad.

Este primer trabajo fue extendido por [Gardey et al. \(2022\)](#), en un trabajo que propone un puntaje de esfuerzo de interacción como un factor que contribuye a la medida de la UX de una página web completa. Este esfuerzo de interacción se calcula automáticamente como una agregación del esfuerzo en cada widget interactivo de una página, para todos los usuarios que han interactuado con ellos.

El esfuerzo en cada widget se predice a partir de diferentes micro-medidas calculadas sobre la interacción del usuario, aprendiendo de las calificaciones manuales de expertos en UX. El documento describe la evaluación del esfuerzo de interacción de diferentes formularios web y cómo se compara con otras métricas de usabilidad e interacción del usuario. También muestra aplicaciones potenciales del puntaje de esfuerzo de interacción en la evaluación automática de páginas web.

Los autores destacan la falta de modelos de evaluación automáticos para medir la UX de sistemas en línea, especialmente en relación con la interacción del usuario. A través de experimentos controlados como pruebas A/B, proponen el concepto de "esfuerzo de interacción" como una métrica que podría usarse para comparar diferentes diseños en la evaluación automática de UX. Este esfuerzo de interacción se ha definido previamente como una puntuación que un experto en UX asigna a la interacción del usuario con un elemento web o widget específico, y puede predecirse a partir de micro-medidas capturadas automáticamente mientras un usuario interactúa con una página web.

En el desarrollo de nuestra investigación sobre el esfuerzo de interacción en dispositivos móviles, hemos considerado importantes contribuciones en el campo de la usabilidad y

medición de la interacción en interfaces. En particular, el estudio de [Gardey et al \(2022b\)](#) se alinea con nuestros objetivos al ofrecer un enfoque novedoso para medir el esfuerzo de interacción en widgets de formularios web. Este trabajo destaca por su metodología para predecir el esfuerzo de interacción a través de micro-medidas automatizadas, basadas en calificaciones de expertos en experiencia de usuario (UX). Esta aproximación es la que debería completarse en un futuro con la base que se crea en esta tesis con las métricas para entrenar un modelo de predicción.

Como los proyectos sólo se desarrollaron para trabajar con computadoras de escritorio desarrollamos el nuestro que tiene en cuenta a la base de usuarios de dispositivos móviles que representa una porción tan enorme del mercado.

Capítulo 3

Desarrollo de Métricas para Dispositivos Móviles

Nuestra propuesta consiste en crear una manera de medir el esfuerzo de interacción en dispositivos móviles. Si bien ya existe una línea de investigación sobre este tema ([Grigera et al. \(2019\)](#)) la misma está centrada en interacción sobre computadoras de escritorio, donde se desarrollaron y entrenaron modelos predictivos que replican la evaluación de un experto en UX sobre el esfuerzo requerido por los usuarios para completar tareas específicas en cada widget. Sin embargo, como se mencionó en la introducción, las aplicaciones móviles presentan un panorama distinto al de las aplicaciones web de escritorio. El número de dispositivos móviles supera al de la población mundial, a diferencia de la cantidad de computadoras personales, lo que resalta la necesidad de realizar estudios específicos para este contexto.

Por este motivo nos propusimos crear un conjunto de métricas que representen la interacción en dispositivos móviles tomando como base lo hecho para la versión de escritorio.

El listado inicial de métricas que hicimos se fundamentó principalmente en los trabajos de Grigera, Gardey et al. explicados en la sección [Contexto 2.2](#). El resto de los trabajos relacionados si bien se basan en mejorar la usabilidad en dispositivos móviles, lo hacen desde contextos muy distintos, por lo que traer métricas de estos a nuestro contexto no es suficientemente relevante.

El proceso completo de adaptación a dispositivos móviles implica considerar diversos aspectos, desde la optimización de la visualización de formularios diseñados originalmente para escritorio hasta la implementación de métodos eficientes para rastrear las interacciones de los usuarios en estos formularios. Además, es necesario incorporar métricas específicas que sean relevantes únicamente en el contexto móvil.

3.1 Adaptación de métricas originales

Para crear la evaluación correcta del esfuerzo en dispositivos móviles nos basamos en el trabajo previamente desarrollado por [Gardey et al. \(2022\)](#), en el que inicialmente se desarrollaron métricas funcionales exclusivamente diseñadas para su implementación en entornos de escritorio, llevamos a cabo ajustes con el objetivo de adecuar dichas métricas para su utilización en dispositivos móviles.

Algunas de estas métricas se mantuvieron bastante similares a las versiones de escritorio, mientras que otras fueron diseñadas específicamente para satisfacer las necesidades únicas de la experiencia móvil.

TactilTypingVariance: La métrica de varianza entre intervalos se encarga de calcular la variabilidad en la velocidad de tipeo a medida que cada letra es introducida. En otras palabras, si al escribir un nombre se experimenta una velocidad inicial lenta seguida de una aceleración notable al final, la varianza entre los intervalos será considerablemente alta, lo que se traduce en un resultado más amplio. En contraste, si la velocidad de escritura se mantiene constante, sin variaciones significativas, la varianza resultante será menor, indicando una coherencia en la velocidad de tipeo a lo largo del proceso. En esencia, esta métrica proporciona una medida cuantitativa de la consistencia o variabilidad en la velocidad de escritura durante la introducción de cada letra.

TactilDwellTime: Es la cantidad de tiempo que el usuario mantiene apoyado el dedo en una entrada del formulario, ya sea una entrada de texto, una selección múltiple, un selección de fecha, etc.

TouchInteractions: La cantidad de veces que el usuario interactúa con cada entrada del formulario

DistanceMoved: Detecta la distancia total recorrida con el dedo apoyado en la pantalla.

MisInteractions: Cantidad de toques erróneos. Los toques que se hacen cerca de cualquier entrada del formulario lo consideramos como un toque erróneo, aunque en el 100% de los casos no signifique eso, es una medida bastante representativa.

TouchVelocity: La velocidad a la que el usuario se desplaza por la pantalla, los rangos van de 0 a 1, cuanto más cerca de 1 más rápido son y cuanto más cerca de 0 más lento.

FocusTimeTactil: La cantidad de tiempo que paso el usuario en cada entrada del formulario.

InputSwitch: Se encarga de detectar alteraciones en el tipo de entrada mientras se completa un input de tipo texto. Más concretamente, esta métrica evalúa si, estando dentro del campo (input), se produce un movimiento fuera del mismo, esto se considera como un cambio en la entrada. Asimismo, si se reanuda la escritura en el campo de entrada después de un evento en el que el usuario se movió fuera del mismo, también se registra como un nuevo cambio de entrada.

En términos más simples, "inputSwitch" detecta modificaciones en el flujo de entrada, ya sea al moverse fuera del input mientras se está adentro o al reanudar la escritura después de haberse salido.

SelectOptionsDisplayTimeTactil: La cantidad de tiempo que se estuvo mostrando en pantalla el selector múltiple hasta que el usuario selecciona alguna opción.

3.2 Evaluación con Participantes

Una vez que implementamos las métricas, era esencial validarlas en un entorno real para evaluar su relación con el esfuerzo de interacción móvil.

Para llevar a cabo esta evaluación, se ideó un método en el que los participantes completaron una serie de formularios disponibles en una página web utilizando sus dispositivos móviles, mientras eran grabados en video. Esta grabación nos permitió analizar detenidamente los videos, observando los movimientos, las pausas y cualquier otra acción realizada por los usuarios. A partir de esta observación, validamos nuestra perspectiva sobre las posibles métricas que debería incluir el programa.

Para llevar a cabo la grabación de cada usuario nos encontramos limitados técnicamente a hacerlo en una misma red local, mediante la cuál una computadora que contará con los requisitos necesarios permitiría que el usuario accediera a una página local donde pudiera visualizar y completar los formularios.

Luego de analizar estas grabaciones, encontramos que las métricas calculadas hasta el momento eran suficientemente adecuadas.

3.2.1 Preparación

Para el trabajo previo sobre interfaces de escritorio, los autores prepararon una serie de aplicaciones web sobre las cuales se ejecutaron experimentos con participantes. Nuestra idea fue la de aprovechar esta infraestructura, pero para esto resultó necesario adaptar las aplicaciones para dispositivos móviles, revistando que cumplan con las prácticas de *responsiveness*.

La infraestructura contaba con 5 formularios de prueba distintos, también con un panel de administración o *backend* en el cual había 15 tablas, las consisten principalmente en registrar las sesiones de los usuarios, una grabación de pantalla y una colección de registros de interacción que contienen las micromedidas correspondientes para cada widget con el usuario que interactuó.

El frontend contaba con un “grabador de pantalla” el cual se encargaba de recopilar la información de los eventos del usuario. En este módulo se encuentra desarrollado el tratamiento de los eventos, o sea las micrométricas. Por otra parte el backend era un SQLite, el cual tiene varias tablas, pero nosotros no utilizamos ninguna de estas, simplemente creamos las necesarias para nuestro uso.

Utilizamos los formularios que ya estaban desarrollados como sitios de pruebas para ir testeando las métricas nombradas en la sección 3.1, los cuales pedían distintos datos al usuario, algunos para sacar un turno ficticio, otros para hacer una compra, etc.

Sabiendo esto, lo primero que tuvimos que hacer fue crear un menú para navegar fácilmente entre los formularios de prueba (ya que se podía acceder mediante URL mas no navegando por el frontend). Luego tuvimos que adaptar la visualización de los formularios de prueba, ya que estos estaban hechos para un contexto de escritorio, por lo que tuvimos que hacer que se vean bien en un dispositivo móvil.

Posteriormente, continuamos hacia la esencia del proyecto, centrándonos en la adaptación de las métricas existentes y las nuevas. En este contexto, seleccionamos y ajustamos las métricas más relevantes, excluyendo aquellas que consideramos más triviales en el contexto de la utilización en dispositivos móviles y agregando nuevas que proporcionen información valiosa al entorno móvil.

Una vez desarrolladas las métricas de la [sección 3.1](#), teníamos que poder almacenar esta información en algún lugar, por lo que optamos por usar sqlite para guardarla.

Y por último una vez tenemos la información el último paso es exportar esto en un archivo csv, el cual servirá en un futuro para entrenar un modelo de predicción.

Detalles específicos sobre la implementación

Como primera medida creamos un menú de navegación para los formularios de prueba, que puede verse en la [Figura 1](#).

The image shows a mobile application interface. On the left, there is a vertical menu with five blue buttons: 'Pasajero', 'Pasaporte', 'Checkout', 'Prestamo', and 'Registrarme'. A red arrow points from the 'Pasajero' button to the right. On the right, there is a form titled 'Datos del pasajero'. The form has a header with 'Grabar' and two buttons: 'Iniciar' (green) and 'Abandonar' (grey). Below the header, there are several input fields: 'Nombre' (text input), 'Apellido' (text input), 'Fecha de Nacimiento' (three dropdown menus for 'Dia', 'Mes', and 'Año'), 'Sexo' (radio buttons for 'Masculino' and 'Femenino'), and 'Tipo de Documento' (text input).

Figura 1. Captura de pantalla del menú principal y derivado

Sobre la izquierda de la Figura 1 podemos ver el menú, y por ejemplo al clicar en Pasajero, nos redirige al formulario correspondiente.

Después de establecer este método de navegación, nos enfocamos en adaptar los formularios para que se vean correctamente en un dispositivo móvil.

Para esto creamos una clase CSS la cual se encarga de cambiar el tamaño de las columnas de los formularios para que abarquen la totalidad del ancho de la pantalla, en la [Figura 2](#) se ve el formulario en su estado inicial.

Volver

Grabar

Iniciar Abandonar

Datos del pasajero

Nombre

Apellido

Fecha de Nacimiento

Día
 Mes
 Año

Sexo

Masculino
 Femenino

Tipo de Documento de

DNI

Numero Documento

Pais de Emisión

Seleccionar

Figura 2. Captura de pantalla de formulario previo a arreglos del CSS

La imagen de la [Figura 3](#) muestra cómo quedan los formularios una vez aplicados los cambios

Volver

Grabar

Iniciar Abandonar

Datos del pasajero

Nombre

Apellido

Fecha de Nacimiento

Día
 Mes
 Año

Sexo

Masculino
 Femenino

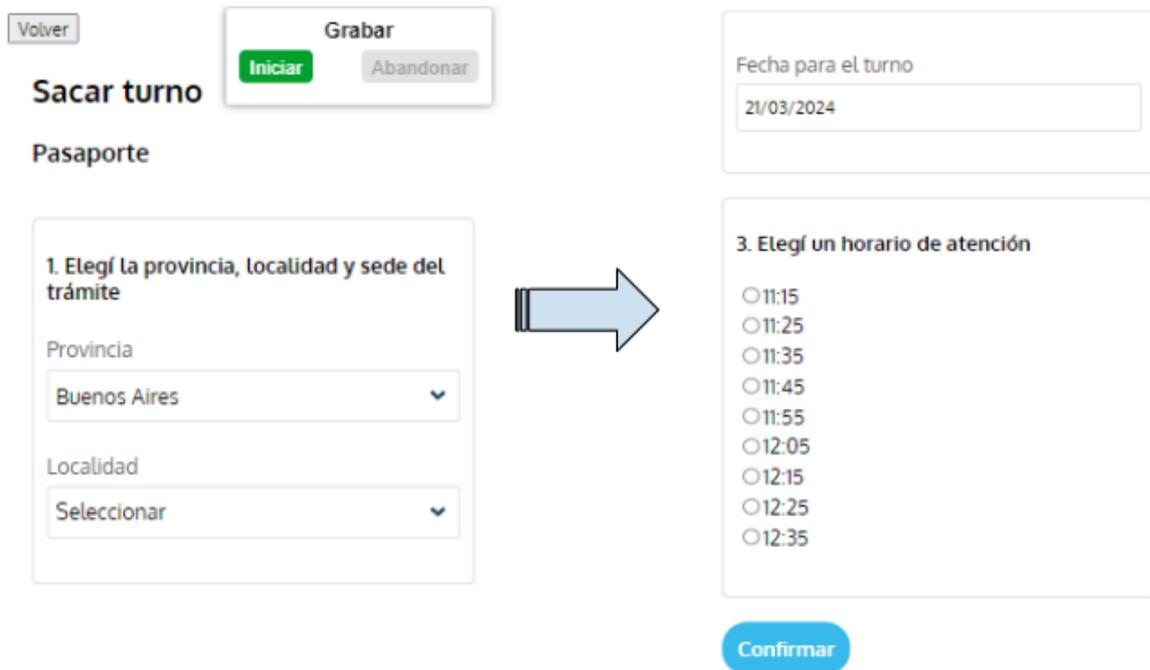
Figura 3 Captura de pantalla de formulario después de aplicar arreglos del CSS

En la [Figura 4](#) podemos ver cómo quedó el formulario para obtener turnos de gestión de

pasaporte, en la Figura 5 el de Checkout, en la Figura 6 la de Préstamo, y en la Figura 7 se observa el Registro.

Pasaporte

Este formulario intenta representar un formulario para sacar un pasaporte. El participante tiene que seleccionar la provincia y localidad del trámite, la fecha y el horario del turno.



El formulario muestra un flujo de trabajo para solicitar un turno de pasaporte. En la parte superior izquierda, hay un botón 'Volver' y un grupo de botones 'Grabar' que incluye 'Iniciar' (en verde) y 'Abandonar'. El título principal es 'Sacar turno Pasaporte'. A la derecha, un campo 'Fecha para el turno' muestra '21/03/2024'. El primer paso, '1. Elegí la provincia, localidad y sede del trámite', contiene dos selectores: 'Provincia' con 'Buenos Aires' seleccionado y 'Localidad' con 'Seleccionar' seleccionado. Una flecha azul apunta a la derecha, indicando el siguiente paso: '3. Elegí un horario de atención', que presenta una lista de horarios con botones de radio: 11:15, 11:25, 11:35, 11:45, 11:55, 12:05, 12:15, 12:25, y 12:35. En la parte inferior derecha, hay un botón azul 'Confirmar'.

Figura 4. Capturas de pantalla del menú Pasaporte

Checkout

Este formulario intenta representar un formulario para la compra de una prenda de vestir. El participante completa datos del envío, como: nombre, dni, correo electrónico y dirección completa. Y por otra parte el método de pago, seleccionando si va a ser por transferencia, mercadopago o tarjeta de crédito, y completando los datos de la tarjeta de crédito en este caso. Se les indicó a los participantes que paguen con tarjeta de crédito para que completaran todos los inputs, y así obtener más información. También les dijimos que completen con datos falsos, y no con su tarjeta de crédito real.

Finalizar Compra

Volver Grabar Iniciar Abandonar

Datos de envío

Nombre

DNI

Correo Electrónico

Método de Pago

Transferencia Bancaria

MercadoPago

Tarjeta de Crédito

Nombre

Número de Tarjeta

16 números

Vencimiento

MM/YYYY

Código

XXX

Figura 5. Capturas de pantalla de menú Checkout

Prestamo

Este formulario representa un formulario para sacar un préstamo. El participante completa el tipo del préstamo (total o en cuotas), el monto del préstamo, el plazo en el cual pagar, tipo de cliente (exclusivo, premium, general, start), y jurisdicción. Les dijimos a los participantes que no importaba lo que completaran, con cualquier dato estaba bien.

Simulá tu préstamo en pesos

Tipo de cálculo
Seleccionar

Monto del Préstamo

Plazo
Seleccionar

Tipo de cliente
Seleccionar

Jurisdicción
Seleccionar

Simular

Figura 6. Capturas de pantalla de menú Préstamo

Registrarme

Este formulario representa un formulario para registrar un usuario. El participante completa con su nombre y apellido, email, contraseña, teléfono, documento, género, repetir mail, repetir contraseña y fecha de nacimiento. Les dijimos a los participantes que completen con contraseñas y emails sencillos, para que no se les complique repetirlos.



La imagen muestra dos versiones de un formulario de registro. A la izquierda, la versión de escritorio, con campos de texto para: Nombre y Apellido, Email, Contraseña, Telefono, and Documento. A la derecha, la versión móvil, con campos para: Genero (menú desplegable con 'No Informar'), Repetir Email, Repetir Contraseña, Fecha de Nacimiento, and un botón azul 'Registrarme'. Una flecha azul apunta de la versión escritorio a la versión móvil.

Figura 7. Capturas de pantalla de menú Registrarme

Luego de estos pequeños cambios vino una de las partes más complejas: el desarrollo de las métricas. Inicialmente, el enfoque del proyecto fue tomar como base la versión de escritorio existente. A partir de ahí, se inició un proceso de adaptación y desarrollo de funciones específicamente diseñadas para dispositivos móviles. Este proceso no es trivial, ya que implica comprender y respetar las diferencias en la interacción del usuario, el tamaño de la pantalla, el tacto y otras peculiaridades de los dispositivos móviles.

En este contexto, partiendo sobre lo heredado de la versión de desktop, desarrollamos un script el cual contiene las métricas, y es el responsable de detectar y registrar las interacciones del usuario con diferentes elementos de la página, como formularios y widgets. El archivo en cuestión se trata de *MicroMetricLogger.js*, y el mismo es capaz de identificar distintos elementos del DOM (Modelo de Objetos del Documento) de la página y monitorear eventos específicos, como clicks, entradas de texto, y selecciones de opciones.

Entonces la atención se dirigió hacia la adaptación y refinamiento de las métricas en el archivo *MicroMetricLogger.js*. Este proceso implicó asegurarse de que las métricas recopiladas fueran igualmente relevantes y precisas para las interacciones en dispositivos móviles, lo cual podría incluir ajustes en la forma en que se detectan y registran las interacciones, dada la variabilidad de los comportamientos de los usuarios en diferentes dispositivos.

A continuación veremos las diferencias entre las micrométricas de desktop que heredamos y como las adaptamos para su versión mobile

Diferencias con la versión desktop

TactilTypingVariance: Para esta micrométrica cambiamos el evento de escucha de los widgets de tipo texto de keypress a keyup. Este cambio es relativamente menor, pero esencial para capturar la variabilidad en la entrada de texto táctil. Implementamos esto mediante el siguiente código:

```
setUp() {  
  addEventListener("input[widget-type='text']", "keyup", this.keyPressHandler);  
  addEventListener("input[widget-type='text']", "blur", this.blurHandler);  
}
```

El resto del código calcula la desviación estándar, sigue siendo efectivo sin cambios, lo que sugiere que los patrones de entrada táctil se comportan de manera consistente con las entradas de teclado físico, código del cálculo de la desviación estándar:

```
standardDeviation(typingIntervals) {  
  if (typingIntervals.length == 0) {  
    return null;  
  }  
  var total = 0;  
  var total_power_of_two = 0;  
  for (var i = 0; i < typingIntervals.length; i++) {  
    total += typingIntervals[i];  
    total_power_of_two += Math.pow(typingIntervals[i], 2);  
  }  
  var media = total / typingIntervals.length;  
  var variance =  
    total_power_of_two / typingIntervals.length - Math.pow(media, 2);  
  return Math.sqrt(variance);  
}
```

Hace el cálculo basándose en los intervalos de tiempo entre los toques obtenidos.

TactilDwellTime: En esta micrométrica nos enfrentamos al desafío de medir el tiempo que un usuario pasa con su dedo sobre un elemento interactivo. Reemplazamos el seguimiento del movimiento del mouse por la captura de los eventos *touchstart* y *touchend*. Este cambio refleja las interacciones táctiles únicas de los dispositivos móviles.

Detectamos cuando se empezaba a tocar la pantalla y cuando se terminaba de tocar, con los eventos "touchstart" y "touchend":

```
addEventListener(this.targetElements, "touchstart", this.handleTouchStart);
addEventListener(this.targetElements, "touchend", this.handleTouchEnd);
```

Luego usando `Date.now()` para obtener la hora en la que se tocó cada evento lo que hicimos fue restar estos dos tiempos para obtener la cantidad que se estuvo presionando el tactil en ese widget. En el evento de cuando se termina de tocar la pantalla se calcula y se guarda la métrica en el widget correspondiente:

```
handleTouchEnd(event) {
  this.endTime = Date.now();
  let tiempoDeDwell = this.endTime - this.startTime;
  this.microMetricLogger.getWidgetLogs(event.currentTarget).tactilDwellTime +=
  tiempoDeDwell;
}
```

TouchInteractions: En la versión de desktop se sumaban las interacciones cuando se clickea un widget, cuando se hace foco en él, y en el caso de los selects cuando se abrían, como se puede ver en el siguiente código:

```
setUp() {
  addEventListener(this.targetElementsSelector, 'focus', this.focusHandler);
  addEventListener(this.linksSelector, 'click', this.clickHandler);
  addEventListener(
    "div[widget-type='select'], div[widget-type='date-select']",
    'open',
    this.onSelectClick
  );
}
```

En nuestro caso simplificamos el proceso. En lugar de sumar las interacciones basadas en eventos de enfoque y click, incrementamos el contador de interacciones cada vez que se detecta un touchstart (un toque):

```
setUp() {
  addEventListener(this.targetElements, "touchstart", this.onInputTouch);
}

onInputTouch(event) {
  this.microMetricLogger.getWidgetLogs(
    event.currentTarget
  ).touchInteractions += 1;
}
```

```
}
```

Este enfoque nos permite capturar con precisión la cantidad de veces que un usuario interactúa con un widget en dispositivos móviles

DistanceMoved: Esta métrica no existe en la versión de desktop. Nosotros la usamos solo para el widget “document” que representa eventos sobre la pantalla que no sean widgets. Se usó el evento “touchMove” y cada vez que se dispara el evento se suma a la métrica.

```
onPageMove(event) {  
  this.microMetricLogger.getWidgetLogs(  
    event.currentTarget  
  ).distanceMoved += 1;  
}
```

MissedInteractions: Esta métrica la adaptamos de la versión de desktop. Al igual que la anterior solo se usa en el widget “document”. Simplemente usamos la función “touchstart” para que cada vez que se toque se ejecute esta función:

```
onTouchPage(event) {  
  let anchors = document.querySelectorAll(this.targetElementsSelector);  
  for (let anchor of anchors) {  
    let clientX = event.changedTouches[0].clientX;  
    let clientY = event.changedTouches[0].clientY;  
    if (  
      !anchor  
        .getBoundingClientRect()  
        .includesPoint(clientX, clientY) &&  
      anchor  
        .getWidgetSurroundings()  
        .includesPoint(clientX, clientY)  
    ){  
      this.microMetricLogger.getWidgetLogs(  
        event.currentTarget  
      ).missedInteractions += 1;  
    }  
  }  
}
```

La cual incrementa en 1 missedInteractions si se toca en un radio de 20 pixeles cerca de un widget.

FocusTimeTactil: Esta métrica en su versión desktop solamente usaba el evento “focus” para detectar cuando se ingresaba a un input y el evento “blur” cuando se salía del input, pero en nuestro caso esos eventos para los widgets de tipo select no nos funcionaba, por lo que tuvimos que cambiar “focus” por “click”, y “blur” por “change”, para los tipo select. De esta manera:

```
setUp() {
  const elements = document.querySelectorAll(this.targetElements);

  elements.forEach((element) => {
    if (element.getAttribute('widget-type') === 'select' || element.getAttribute('widget-type') ===
'date-select') {
      element.addEventListener('click', this.focusHandler);
    } else {
      element.addEventListener('focus', this.focusHandler);
    }
  });

  elements.forEach((element) => {
    if (element.getAttribute('widget-type') === 'select' || element.getAttribute('widget-type') ===
'date-select') {
      element.addEventListener('change', this.blurHandler);
    } else {
      element.addEventListener('blur', this.blurHandler);
    }
  });
}
```

La manera en la que funciona es que cuando se entra al input se toma el tiempo actual y se resta al tiempo actual cuando se sale, y de esa forma se calcula el tiempo que se estuvo enfocando en el input:

```
blurHandler(event) {
  this.microMetricLogger.getWidgetLogs(event.target).enteredText =
  event.target.value;
  if (!this.currentWidget) {
    return null;
  }
  this.blurTime = event.timeStamp;
  this.focusTime = this.blurTime - this.startTime;
  this.logFocusTime(event);
}
```

LogFocusTime se encarga de guardar el focusTime en su respectivo widget.

InputSwitch: Esta métrica prácticamente no la modificamos en comparación con la versión de desktop, solamente se modificaron los eventos por sus equivalentes a la versión mobile. Se modificó "mousemove" por "touchmove" y "keypress" por "keyup". El funcionamiento de la métrica consta de detectar cambios en el tipo de interacción, se guarda la última interacción y si la misma cambia se aumenta el "inputSwitch" en 1:

```
keypressHandler(event) {
  if (this.lastAction == "touchmove") {
    this.microMetricLogger.getWidgetLogs(this.currentWidget).inputSwitches++;
  }
  this.lastAction = "keyup";
}

touchMoveHandler(event) {
  if (this.lastAction == "keyup") {
    this.microMetricLogger.getWidgetLogs(this.currentWidget).inputSwitches++;
  }
  this.lastAction = "touchmove";
}
```

SelectOptionsDisplayTimeTactil: Para esta métrica solamente se modificó el evento "click" por "touchstart". La manera en la que funciona es que cuando se toca el select se empieza a contar el tiempo y cuando sucede el evento "change", es decir, cuando se selecciona una opción, se guarda el tiempo en ese momento.

```
onChange(event) {
  this.microMetricLogger.getWidgetLogs(event.target).optionsDisplayTime +=
    this.getOptionsDisplayTime(event);
  if (event.target.getAttribute("widget-type") == "date-select") {
    this.microMetricLogger.getWidgetLogs(event.target).optionsCount =
      this.microMetricLogger
        .getDateSelectNamed(event.target.getAttribute("data-select-name"))
        .getTotalOptionsCount();
  } else {
    this.microMetricLogger.getWidgetLogs(event.target).optionsCount =
      event.target.getOptionsCount();
  }
}
```

Además en la función *onChange* se guarda la cantidad de opciones que tiene el select, en *optionsCount*.

TouchVelocity: Esta métrica calcula la media de la velocidad a la que se arrastra el dedo sobre la pantalla en píxeles por milisegundo. Lo hace de la siguiente manera:

```
handleTouchMove(event) {
  if (this.startX !== null && this.startY !== null) {
    const currentX = event.touches[0].clientX;
    const currentY = event.touches[0].clientY;
    const distanceX = currentX - this.startX;
    const distanceY = currentY - this.startY;
    const distance = Math.sqrt(distanceX * distanceX + distanceY * distanceY);
    const currentTime = Date.now();
    const timeElapsed = currentTime - this.startTimeVelocity;

    // Calcular la velocidad en píxeles por milisegundo
    const velocity = distance / timeElapsed;
    this.velocityIntervals.push(velocity);
    let velocityMedia = this.calculateMedia(this.velocityIntervals);
    // Registra la métrica de velocidad en píxeles por milisegundo
    this.microMetricLogger.getWidgetLogs(
      event.currentTarget
    ).touchVelocity = velocityMedia;
    // Actualiza la posición inicial y el tiempo
    this.startX = currentX;
    this.startY = currentY;
    this.startTimeVelocity = currentTime;
  }
}

calculateMedia(typingIntervals) {
  if (typingIntervals.length == 0) {
    return null;
  }
  var total = 0;
  for (var i = 0; i < typingIntervals.length; i++) {
    total += typingIntervals[i];
  }
  var media = total / typingIntervals.length;
  return media;
}
```

Calcula la distancia en píxeles que se ha movido el dedo en los ejes X e Y desde el punto inicial del movimiento táctil. Utiliza estas distancias para calcular la distancia total utilizando el teorema de Pitágoras.

Luego calcula la velocidad dividiendo la distancia total recorrida por el tiempo transcurrido desde el último cálculo de velocidad. El tiempo se mide en milisegundos.

Y por último se agrega la velocidad a un array para calcular la media entre todos.

Backend

Como se dijo anteriormente, contamos con un backend previamente hecho en la versión desktop, el cual consistía de 15 tablas, de las cuales no usamos ninguna, ya que creamos las que nosotros necesitábamos. Al no tener nosotros usuarios, como sí tenía la versión en desktop, no tenían utilidad para nosotros varias tablas relacionadas con esto.

El backend estaba desarrollado en Python con Django DB, y nosotros decidimos usar lo mismo. Se ha adoptado un enfoque de diseño donde se crea un modelo dedicado para cada widget. Este enfoque permite almacenar de manera sistemática cada interacción realizada en los formularios asociados a esos widgets. En el contexto de este diseño, cada campo de las tablas de la base de datos refleja las micrométricas específicas del widget al que está vinculada.

Se ha utilizado la funcionalidad de migraciones proporcionada por Django para gestionar la creación y evolución de las tablas de la base de datos, asegurando así la coherencia y la integridad de los datos almacenados en relación con las estructuras de los modelos de los widgets.

Modelado de Datos

Para gestionar la información recopilada a través de los distintos widgets de la interfaz de usuario, se diseñó un conjunto de modelos dedicados. Cada modelo corresponde a un tipo de widget específico y almacena las micrométricas relevantes asociadas a las interacciones del usuario con dicho widget. Los campos de cada modelo reflejan las características particulares de las interacciones, como el tiempo de permanencia táctil, la variación en la escritura, el número de interacciones táctiles, entre otros.

La estructura de los modelos se puede resumir de la siguiente manera:

TextInputMetrics: Almacena métricas relacionadas con la interacción del usuario con campos de texto, como el tiempo de enfoque, la variación en la escritura y el texto ingresado.

SelectInputMetrics: Registra métricas de campos de selección, incluyendo el número de opciones y el tiempo de visualización de las opciones.

RadiInputMetrics: Guarda información sobre la interacción con botones de radio, como el número de opciones y el tiempo de enfoque.

DateSelectInputMetrics: Similar a SelectInputMetrics, pero específico para selecciones de fecha.

DocumentMetrics: Captura métricas relacionadas con la interacción con documentos, como la distancia recorrida por el toque y la velocidad del toque.

DatePickerInputMetrics: Almacena información específica de la interacción con selectores de fecha.

La implementación de estos modelos se realiza utilizando la sintaxis de Django DB, con campos como CharField, IntegerField, FloatField y BigIntegerField para representar los diferentes tipos de datos.

Gestión de Migraciones

Para manejar la creación y evolución de las tablas de la base de datos correspondientes a estos modelos, se utilizó el sistema de migraciones de Django. Este enfoque asegura la coherencia y la integridad de los datos a lo largo del tiempo, permitiendo adaptar la estructura de la base de datos a medida que evolucionan los requisitos de la aplicación.

En la [Figura 8](#) se puede observar el diagrama de las entidades creadas

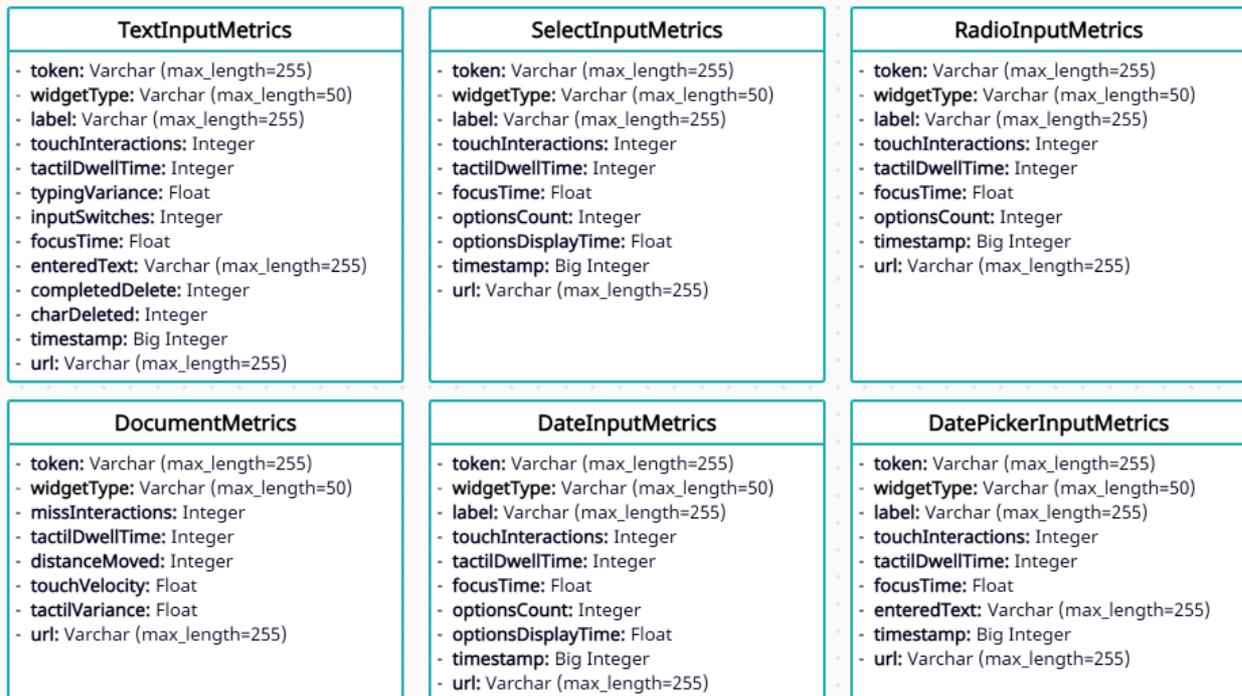


Figura 8. Diagrama de entidades

Siempre que un usuario termina de completar un formulario se envía la información al endpoint “training_models/sendData” donde se tratan los datos y se insertan en sus respectivas tablas.

Básicamente el código recorre el diccionario que tiene las micrométricas y va filtrando a que widget corresponde, para así poder almacenar la información en su respectiva tabla.

A continuación se puede ver la lógica con la que se fue filtrando la información para ser almacenada:

```
token = request_data.get("token")
data = request_data.get("data") // Conjunto de datos recopilados
for key, item in data.items(): // Se recorre el conjunto de datos
    item["token"] = token

    if item["widgetType"] == "TextInput":
        nuevo_registro = models.TextInputMetrics(
            token=item["token"],
            widgetType=item["widgetType"],
            label=item["label"],
            touchInteractions=item["touchInteractions"],
```

```

    tactilDwellTime=item["tactilDwellTime"],
    typingVariance=item["typingVariance"],
    inputSwitches=item["inputSwitches"],
    focusTime=item["focusTime"],
    enteredText=item["enteredText"],
    completedDelete = item["completedDelete"],
    charDeleted = item["charDeleted"],
    timestamp=item["timestamp"],
    url = item["url"]
)
nuevo_registro.save()
elif item["widgetType"] == "SelectInput":
    nuevo_registro = models.SelectInputMetrics(
        token=item["token"],
        widgetType=item["widgetType"],
        label=item["label"],
        touchInteractions=item["touchInteractions"],
        tactilDwellTime=item["tactilDwellTime"],
        focusTime=item["focusTime"],
        timestamp=item["timestamp"],
        optionsCount=item["optionsCount"],
        optionsDisplayTime=item["optionsDisplayTime"],
        url = item["url"]
    )
    nuevo_registro.save()

```

De la misma manera se hicieron *ifs* con cada widget para ir filtrando y guardando la información en su tabla correspondiente.

Una vez conseguida toda la información programamos un pequeño script en Python el cual exporta las tablas de la base de datos y las transforma en archivos CSV, para luego usarlos en un modelo de predicción en posibles trabajos futuros.

Para cada prueba realizada se tomaban los resultados de métricas de ese usuario para hacer los CSV y adicionalmente un video donde se viera por completo cómo interactuaban con los formularios e iban completando lo requerido para hacer una evaluación del esfuerzo que le lleva a cada uno. Estos son los videos que los expertos en UX analizaron en la etapa posterior para poder encontrar posibles métricas que faltaran a las creadas al inicio, como también cuales pueden llegar a ser las valiosas a la hora de implementarlas en un modelo de predicción.

El siguiente código fue el utilizado para la exportación a CSV de los datos obtenidos, la cual crea varios archivos CSV, uno por cada widget:

```

def exportar_a_csv(tabla, archivo_csv):
    # Conectar a la base de datos SQLite
    conn = sqlite3.connect('db.sqlite3')
    cursor = conn.cursor()

    # Consulta SQL para seleccionar todos los datos de la tabla
    consulta_sql = f'SELECT * FROM {tabla}'

    # Ejecutar la consulta SQL
    cursor.execute(consulta_sql)

    # Obtener los nombres de las columnas
    nombres_columnas = [descripcion[0] for descripcion in cursor.description]

    # Abrir un archivo CSV para escribir
    with open(archivo_csv, 'w', newline='') as archivo:
        writer = csv.writer(archivo)

        # Escribir los nombres de las columnas como la primera fila del archivo CSV
        writer.writerow(nombres_columnas)

        # Escribir los datos de la tabla en el archivo CSV
        for fila in cursor:
            writer.writerow(fila)

    # Cerrar la conexión a la base de datos
    conn.close()

#Lista de tablas a exportar
tablas = ['training_models_textinputmetrics',
          'training_models_datepickerinputmetrics',
          'training_models_dateselectinputmetrics',
          'training_models_radioinputmetrics',
          'training_models_documentmetrics',
          'training_models_selectinputmetrics']
#Exportar cada tabla a un archivo CSV
for tabla in tablas:
    archivo_csv = f'{tabla}.csv'
    exportar_a_csv(tabla, archivo_csv)

```

Infraestructura para presentar las aplicaciones de prueba
Contexto y Objetivo de la Implementación

En el proceso de desarrollar nuestra aplicación nos enfrentamos al desafío de probar eficientemente la aplicación en un entorno real, es decir, en dispositivos móviles. Nuestro objetivo era validar las métricas relacionadas con la experiencia de usuario, las cuales eran cruciales para el éxito del proyecto.

Para realizar las pruebas, necesitábamos acceder a nuestra aplicación desde dispositivos móviles. Hicimos esto ingresando a la aplicación a través del navegador móvil, utilizando la dirección IPv4 de nuestra computadora donde estaba alojada la aplicación. Este método requería que tanto la computadora como el dispositivo móvil estuvieran conectados a la misma red Wi-Fi.

Sin embargo, enfrentamos un obstáculo para acceder al backend de nuestra aplicación desde dispositivos externos, ya que este residía en un entorno de desarrollo local. Aquí es donde NGROK se volvió esencial. NGROK es una herramienta gratuita que nos permitió exponer nuestro servidor local a Internet, creando un túnel seguro. Esto significa que pudimos hacer accesible nuestro backend local en la web, permitiendo que los dispositivos móviles interactúen con él sin la necesidad de alojar la aplicación en un servidor remoto.

Para utilizar NGROK, lo primero fue instalarlo. Una vez instalado, NGROK nos proporcionó un comando en la terminal para exponer cualquier puerto de nuestro entorno local a la web. En el desarrollo de aplicaciones móviles, no es práctico usar 'localhost' como URL en dispositivos reales o emuladores. NGROK nos ofreció una solución a este problema al proporcionarnos URLs únicas y accesibles globalmente. Iniciamos el frontend de la aplicación con `npm start`, también hacemos `npm run dev` para que construya el javascript (para así poder usar los nuevos endpoints), y el backend con `python manage.py runserver 9090`. Luego, ejecutamos `ngrok http 9090` para exponer el puerto 9090. NGROK generó una URL, como `"https://0879-190-246-110-192.ngrok-free.app"`, que usamos en nuestro código para conectar con los endpoints del backend, facilitando operaciones como el login, logout y la transferencia de datos para el CSV.

Una vez configurado NGROK, pudimos acceder a la aplicación desde cualquier navegador móvil ingresando la dirección IPv4 de nuestra computadora y el puerto asignado, como `"http://192.168.0.231:8080/"`. Esta etapa fue crucial para realizar pruebas y asegurar que todas las métricas y funcionalidades funcionaran correctamente en un ambiente real.

Una vez configurado todo lo necesario para llevar a cabo las pruebas, los usuarios deberían completar, uno por uno, todos los formularios de la aplicación mientras son grabados. Para más detalles sobre estas pruebas está la sección [3.2.2 Ejecución](#).

Para concluir cada sesión de prueba, cada vez que un usuario finalizaba la tarea de completar todos los formularios, inmediatamente después, ejecutábamos el script de Python previamente mencionado. Esta acción nos permitía transformar la información recién recopilada del usuario en un archivo CSV estructurado. Este archivo se convertía así en un registro detallado y organizado de los datos de la sesión, facilitando su posterior análisis.

Tras la generación exitosa del archivo CSV, procedíamos a una limpieza de la base de datos para mantener la integridad y la organización de nuestro sistema. Al vaciar la base de datos, nos asegurábamos de que quedará preparada para la siguiente sesión de usuario. Este ciclo de recolección de datos y reinicio sistemático formaba una parte integral de nuestro proceso de prueba, asegurando que cada usuario empezará su experiencia en un entorno limpio.

3.2.2 Ejecución

Para llevar a cabo el estudio se reclutaron 14 voluntarios (8 hombres y 6 mujeres) que eran familiares o conocidos nuestros, los cuales tenían edades comprendidas entre los 15 y 55 años, 34 años en promedio, con una desviación estándar de 14. Para el reclutamiento de estos participantes, se solicitó su colaboración ya que tenían que estar dispuestos a dedicar un tiempo a completar diversos formularios en línea. Las instrucciones proporcionadas a los voluntarios fueron simples:

- 1) Presionar el botón de grabación antes de comenzar a llenar cada formulario en el sitio web designado, el cual contiene 5 formularios para ser completados
- 2) Completar los formularios en el orden que aparecían.

A la hora de completar los formularios se les indicó que podían usar algún dato falso en casos como el de “tarjeta de crédito”, documento, contraseña, y también otros del estilo como fechas para turnos ficticios, básicamente que el usuario se sienta en la libertad de no poner los datos personales que quisiera. Por supuesto siempre respetando la lógica de lo requerido, por ejemplo, no escribir un mail sin arroba, o un número de documento inexistente, etc.

En el caso de los primeros 4 usuarios de prueba completaron todos los formularios menos el de “checkout” porque tenía muchas condiciones en el caso de los números de la tarjeta, que luego fueron arregladas y las siguientes 10 muestras fueron completando todos los formularios que había.

Durante la sesión, se grabó a cada participante mientras completaba los formularios asignados. Los datos resultantes de estas interacciones se registraron y almacenaron en nuestra base de datos, junto con los videos de los participantes en acción, para su análisis posterior.

En la [Figura 9](#) se observa cómo se ven los videos grabados

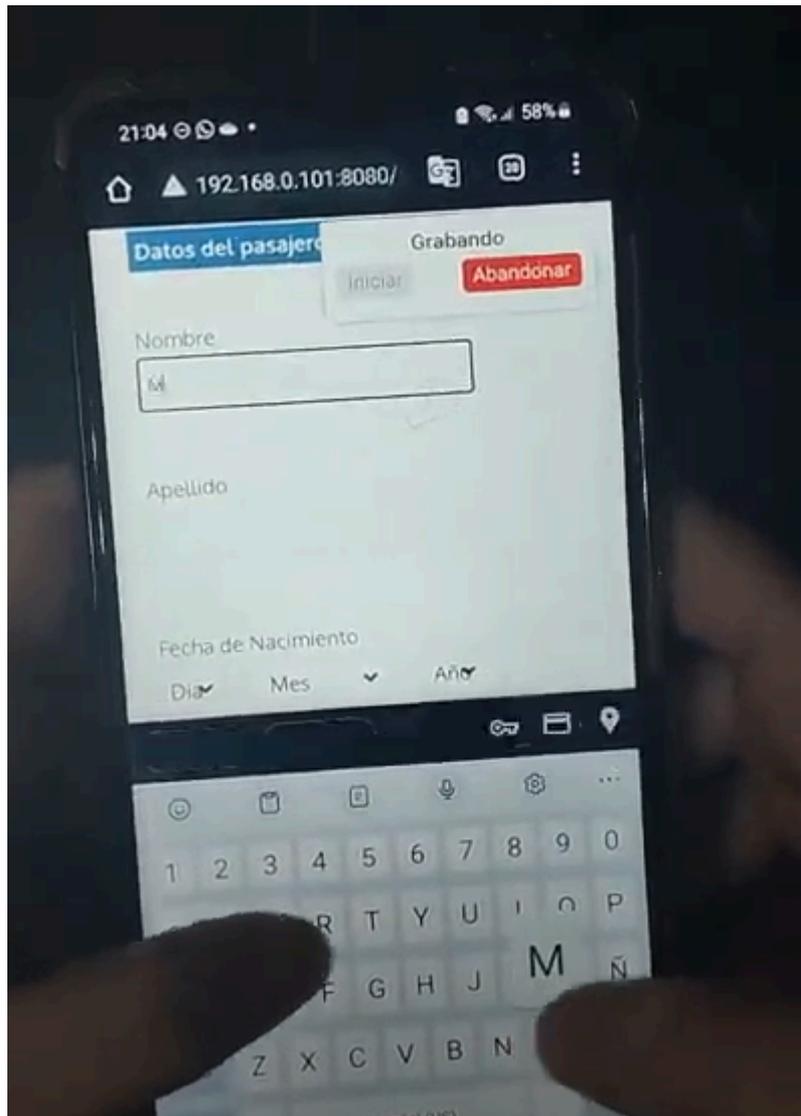


Figura 9. Captura de pantalla de video grabado

Para las pruebas, a algunos participantes se les entregó un dispositivo proporcionado por nosotros, mientras que otros con sus propios teléfonos. En ambos escenarios, colocamos el celular que grabaría la sesión sobre un soporte elevado. Esta disposición permitía capturar una vista aérea clara de las interacciones del usuario con los formularios. Aunque la cercanía del soporte a sus manos podía resultar incómoda, era esencial para obtener una perspectiva detallada de sus acciones.

La tarea asignada a los usuarios consistía en completar, uno por uno, todos los formularios de la aplicación. Cada vez que accedían a un formulario, debían activar manualmente la grabación de sus acciones antes de proceder a llenarlos. Deliberadamente, proporcionamos instrucciones mínimas, enfocándonos en permitir que los usuarios interactúen con los formularios de manera intuitiva. Les indicamos que para información sensible, como datos de tarjetas de crédito, inventaran datos falsos pero memorables.

3.2.3 Resultados

Se detallarán los resultados recopilados en los archivos CSV de las pruebas de los participantes. Los campos son los mostrados en la [sección 3.2.1 backend](#).

Los resultados consisten en un archivo CSV por cada tipo de widget, varios de los campos se repiten entre los distintos widgets, veamos todos los widgets menos el tipo *document* del cual se hablará más adelante ya que es un tanto especial.

widgetType, *label*, *touchInteractions*, *tactilDwellTime*, *focusTime*, *url* y *timestamp* se encuentran en común entre todos los widgets.

widgetType: Se trata de un texto el cual indica que tipo de widget es, por ejemplo *DatePicker*, o *TextInput*.

label: Es el texto de lo que pide el input en cuestión, por ejemplo, Tipo de documento (un input que le pide al usuario ingresar tipo de documento), o Nacionalidad (elegir la nacionalidad).

touchInteractions, **tactilDwellTime**, **focusTime:** Están explicados en la sección 3.1

url: Se trata de la url en el navegador del formulario.

timestamp: Es una marca de tiempo, indica la fecha, hora, minutos y segundos de cuando se hizo la toma de datos.

Luego hay algunos campos específicos para cada tipo de widget, o en común entre algunos pocos, a continuación veremos el resto de campos:

enteredText se encuentra en los *DatePicker* y los *TextInput*, se encarga de guardar el texto de lo que completó el usuario en el caso del *TextInput*, y en el caso del *DatePicker* la fecha que selecciono.

optionsCount se encuentra en los *SelectInput*, *RadioSet* y *DateSelect*, se encarga de guardar la cantidad de opciones que tiene el input para seleccionar.

optionsDisplayTime se encuentra en los *SelectInput* y *DateSelect*, se encarga de guardar el tiempo que se mantuvo abierto el abanico de opciones proporcionado por el widget.

inputSwitches se encuentra en el *TextInput* y se encarga de guardar la cantidad de cambios en el tipo de entrada del widget, por ejemplo, en este caso si uno empieza a escribir y luego sale del input para luego volver a seguir escribiendo, eso contaría como un cambio en el tipo de entrada.

Por último nos queda ver el widget *Document*, el cual como dijimos es especial, ya que no se trata de un input, si no de una entidad para recopilar información en general de la pantalla.

Consta de tres campos específicos, que son: *distanceMoved*, *missedInteractions* y *touchVelocity*, los tres están explicados en la sección 3.1

En la [Figura 10](#) se puede observar un resultado del archivo CSV en formato de tabla sobre el widget *Document*:

<i>widgetType</i>	<i>tactilDwellTime</i>	<i>distanceMoved</i>	<i>missInteractions</i>	<i>touchVelocity</i>	<i>url</i>
Document	6343	98	45	0.03024894203638657	http://192.168.0.231:8080/#/pasajero
Document	1156	7	12	0.021396636924269824	http://192.168.0.231:8080/#/pasaporte
Document	1975	15	18	0.3667332952550648	http://192.168.0.231:8080/#/prestamo
Document	4641	51	31	0.7187214990768586	http://192.168.0.231:8080/#/registrarr

Figura 10. Captura de pantalla de un resultado del tipo de widget *Document*

3.2.4 Amenazas a la validez

Durante la realización de las pruebas, los usuarios fueron observados y grabados mientras completaban los formularios. Sin embargo, la necesidad de sujetar el teléfono fijo con la pantalla hacia arriba, y en algunas pruebas tener que girar un poco la cabeza debido a la presencia de un soporte que obstruía la visión directa al dispositivo, podría haber resultado incómoda y afectar la forma en que interactuaban con el dispositivo.

Los formularios utilizados en las pruebas fueron creados únicamente con fines de prueba, lo que significa que los datos que se les pidió completar no fueron del todo precisos y en su mayoría fueron inventados. Esta falta de datos reales podría haber influenciado la forma en que los usuarios interactuaron con los formularios, ya que la situación simulada puede no reflejar completamente la realidad.

En algunas pruebas, algunos usuarios no utilizaron su propio celular para completar los formularios. Esto podría introducir un sesgo en los resultados, ya que el uso de un dispositivo desconocido podría afectar la familiaridad y comodidad del usuario con el dispositivo, lo que a su vez podría influir en la forma en que completan los formularios.

Durante las pruebas, se proporcionó información a los usuarios para evitar que se complicaran o quedaran trabados con algunos requisitos de los formularios. Esta asistencia podría haber influido en los resultados al facilitar la interacción y reducir la carga cognitiva de los usuarios.

Además, se identificó que la métrica *missedInteractions*, que calcula la cantidad de toques que no se hacen correctamente, puede no ser del todo precisa. En realidad, esta métrica podría estar subestimando la cantidad real de interacciones incorrectas, ya que no tiene en cuenta otros tipos de interacciones incorrectas, como toques mal dirigidos que no llegan a registrarse como interacciones.

3.2.5 Conclusiones

En este capítulo, se abordó el desafío de desarrollar un conjunto de métricas específicas destinadas a evaluar el esfuerzo de interacción en dispositivos móviles, una necesidad que

había sido poco explorada en comparación con la investigación centrada en aplicaciones web de escritorio. Estas métricas, que abarcan aspectos como la variabilidad en la velocidad de tipeo, el tiempo de permanencia táctil y la velocidad de desplazamiento en la pantalla, se adaptaron cuidadosamente y se validaron con éxito para capturar la experiencia de usuario en el contexto móvil, lo que representa una contribución significativa al campo de la experiencia de usuario (UX).

La evaluación con participantes, que incluyó a individuos de diversas edades, reafirmó la relevancia y precisión de estas métricas en un entorno real. Los resultados de esta evaluación destacaron la importancia de considerar una amplia gama de perfiles de usuario, ya que se observaron diferencias notables en la facilidad de completar tareas entre los usuarios más jóvenes y los mayores. Este hallazgo subraya la necesidad de diseñar aplicaciones móviles que sean accesibles y amigables para todos los usuarios, independientemente de su edad o habilidad tecnológica.

Además, la implementación de herramientas como NGROK y la adaptación de la infraestructura existente demostraron ser estrategias efectivas para facilitar las pruebas en dispositivos móviles, permitiendo un acceso remoto al backend y una visualización adecuada de los formularios de prueba en diferentes dispositivos. Estas herramientas y metodologías de desarrollo resultaron ser cruciales para el éxito de la evaluación y proporcionaron un marco sólido para la recolección y análisis de datos.

Los resultados obtenidos y las métricas desarrolladas en este capítulo proporcionan una base sólida para la evaluación de la experiencia de usuario en dispositivos móviles. La adaptación y validación de estas métricas en un entorno real han demostrado su relevancia y utilidad en la captura del esfuerzo de interacción. Además, este trabajo sienta las bases para futuras investigaciones y refinamientos en el campo de la usabilidad móvil.

Nos encontramos con que los usuarios más jóvenes generalmente completaban las tareas con facilidad y pocas complicaciones. Por otro lado, los participantes de 40 años o más enfrentaban más desafíos, y fue precisamente de este grupo de donde surgieron los hallazgos más significativos para nuestras métricas.

A medida que avanzamos en este estudio, el siguiente capítulo se centrará en el refinamiento de estas métricas basándonos en el análisis de expertos en UX, los cuales enriquecerán aún más nuestro conjunto de herramientas de evaluación y nos permitirán abordar de manera más efectiva los desafíos específicos de la interacción del usuario en dispositivos móviles.

Capítulo 4

Refinamiento del Listado de Métricas

Las métricas obtenidas en la evaluación presentada en el capítulo 3 fueron evaluadas por nosotros, por lo que podían no ser del todo representativas. Por esa razón le pedimos a expertos en UX que analicen los videos de la sección 3.2.2, para que nos den su opinión acerca de qué considerar a la hora de medir el esfuerzo.

En este capítulo, presentamos las métricas que los expertos identificaron como cruciales tras revisar los videos con las grabaciones de usuarios completando los formularios que les proporcionamos. Estos materiales les permitieron observar y analizar detenidamente las interacciones de los usuarios, destacando aquellas métricas indispensables para nuestro trabajo. Luego de que los expertos analizaron los videos, se tomaron el tiempo de determinar que tan bien o mal completaron los formularios los usuarios.

4.1 Preparación

Iniciamos el proceso creando un espacio compartido en Google Drive, al que tanto los expertos como nosotros tuvimos acceso. En este espacio, subimos los videos de los usuarios completando los formularios, acompañados de archivos CSV que contenían los resultados de estas pruebas. Los expertos fueron asignados para revisar estos materiales y evaluar el desempeño de los usuarios sobre cada widget en una escala del 1 al 4, donde 1 indicaba el mejor desempeño y 4 el peor. El objetivo era que, a través de los peores desempeños pudiéramos obtener mejores ideas para crear las métricas necesarias o faltantes, ya que, cuanto mejor fuera la puntuación menos errores cometió un usuario más difícil es pensar una métrica para analizar su desempeño al ser “perfecto” o “excelente”

Por ejemplo en la [Figura 11](#) podemos ver cómo puntuó un experto los widgets de cada formulario.

Usuario N°	PASAJERO										PASAPORTE			
	Nombre	Apellido	F.Nac.	Sexo	TDoc	Doc	Pais	Doc	Caducid.	Nac	Prov	Loc	Fecha	Hora
Usuario 6	1	1	2	1	1	1	1	1	1	1	-	2	1	1
Usuario 7	2	1	1	1	1	3	2	2	2	1	1	1	1	1
Usuario 9	1	1	2	1	1	1	1	2	2	2	1	1	1	1

Figura 11 A.

CHECKOUT											PRESTAMO				
Nom	DNI	Email	Dir	Ciud	CP	Pag	Nom	Tarj	Vto	Cod	Tipo C.	Monto	Plazo	Tipo C.	Jurisd
1	1	1	1	1	1	1	1	1	3	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	3	1	1	1	1	1	1
1	1	1	1	1	1	1	4	1	1	1	1	1	1	1	1

Figura 11 B.

REGISTRARME									
NyAp	Email	Contras.	Tel	Doc	Genero	RepEm	RepCo	F.Nac	
1	2	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	2
1	1	1	1	1	1	1	1	1	1

Figura 11 C.

Figura 11. Capturas de pantalla de la planilla con las puntuaciones dadas por el experto en cada widget

Luego de que ellos terminaran ese trabajo, se contactaron con nosotros para acordar una reunión por Google Meet en la que les realizamos una entrevista para que nos explicaran qué aspectos de interacción analizaron para poner dichos puntajes.

La idea de esos intercambios en las reuniones sería que los expertos no tuvieran contexto sobre las métricas que habíamos pensado nosotros para evitar sesgos en su juicio, y de esa manera podrían brindarnos nuevas ideas.

4.2 Ejecución

Organizamos vía correo electrónico una reunión por Google Meet con cada experto, para que nos explicaran en qué se basaron para asignar sus puntuaciones. Durante la reunión, revisamos juntos los videos que habíamos grabado, y a medida que los participantes iban completando los formularios, nos comentaban los factores que tuvieron en cuenta al asignar los puntajes. Esto nos proporcionó una nueva perspectiva sobre qué considerar al definir las métricas significativas para calcular el esfuerzo de los formularios.

Luego de esta reunión con los expertos, nos proporcionaron sus comentarios sobre las posibles métricas a considerar. Durante este proceso, pudimos identificar dos métricas que no habíamos tenido en cuenta inicialmente. Además, observamos que la mayoría de las métricas sugeridas por los expertos ya estaban siendo contempladas en nuestro sistema. Este feedback fue muy útil para afinar y mejorar nuestras métricas, asegurando que cubran todos los aspectos necesarios para una experiencia de usuario óptima.

4.3 Listado Refinado

La lista que sigue no solo enumera cada aspecto considerado importante por los expertos para el desarrollo de estas métricas, sino que también detallamos cómo cada una ya forma parte de nuestro sistema actual o cómo cumplimos con la función que ellos consideran esencial. Además, para aquellas métricas que aún no hemos implementado, ofrecemos una descripción de su funcionalidad prevista y cómo planeamos integrarlas en nuestro sistema para mejorar aún más el análisis y la comprensión de la experiencia del usuario.

Contador de Borrados (a implementar): Los expertos sugieren implementar un contador para registrar el número total de veces que un usuario borra información en los formularios.

Nos parecía necesaria la idea de medir los borrados en los campos de texto, así que decidimos desarrollar una métrica específica para esto. Esta nueva métrica se enfoca en dos cosas: cuenta cuántas veces los usuarios borran completamente un campo de texto y también cuántas veces borran caracteres individuales. Más adelante detallaremos más sobre cómo hemos implementado y utilizado esta métrica en nuestro análisis

Registros de Misclicks (ya implementado): Observar y contabilizar los misclicks (pulsaciones accidentales) en diferentes widgets para entender mejor la usabilidad de la interfaz.

Entendiendo la importancia de esta métrica, hemos adaptado y aplicado un método proveniente de la versión de escritorio de nuestra plataforma. Este enfoque nos permite contabilizar de manera precisa los misclicks en cada widget, acumulando un total que refleja la frecuencia de estas interacciones accidentales.

Selecciones Múltiples en Selects (ya implementado): Monitorear si los usuarios realizan más de una selección en opciones de selección única, lo que puede indicar confusión o problemas de interfaz.

Para monitorear la eficiencia en la selección de opciones utilizamos el método `SelectOptionsDisplayTime`. Este registra el tiempo total empleado en realizar selecciones dentro de un menú desplegable. Por ejemplo, si un usuario tarda 4 segundos en seleccionar "Buenos Aires" y luego otros 4 segundos en elegir una opción diferente, el tiempo total registrado sería de 8 segundos. Adicionalmente, tenemos implementado `focusTime` para medir el tiempo total que el usuario pasa en el select, y `touchInteractions` para contar la cantidad de interacciones con el select.

Velocidad de Completado por Widget (ya implementado): Medir el tiempo que tarda un usuario en completar cada widget, proporcionando información valiosa sobre la eficiencia de la interfaz.

Para esta métrica, nos apoyamos en `focusTime`, que acumula la cantidad de tiempo que un usuario invierte en completar cada widget.

Detección de Borrados Completos (a implementar): Implementar un sistema para detectar cuándo un usuario realiza un borrado completo de un campo de texto.

Este aspecto se integra dentro de nuestra métrica de "Contador de Borrados", abarcando específicamente los casos en que un usuario elimina todo el contenido de un campo de texto de una sola vez. Esta distinción es crucial para entender las diferentes dinámicas de interacción

del usuario con los formularios y nos ayuda a identificar posibles áreas de mejora en la claridad y usabilidad de la interfaz.

Interacciones con Widgets Desplegables (ya implementado): Registrar las veces que se abren y cierran los widgets desplegados, como listas desplegadas, para evaluar su facilidad de uso.

Utilizamos `touchInteractions` para registrar las veces que se interactúa con widgets desplegados. Esto incluye tanto la apertura como la selección de opciones en estos elementos, proporcionando datos sobre su usabilidad.

Cantidad de Correcciones (ignorado): Contar las correcciones hechas por los usuarios, lo que puede indicar áreas de mejora en la claridad de los formularios.

Esta métrica la vemos muy relacionada con la métrica de borrados. Ya que si tenemos la cantidad de borrados de caracteres y totales que se hicieron, tenemos una noción de las correcciones que se hicieron en dicho input.

Demoras entre Toques (a implementar): Medir el tiempo entre interacciones táctiles para entender mejor el flujo de uso y la decisión del usuario.

Se desarrolló la métrica incorporando métodos para medir el tiempo entre interacciones táctiles y así comprender mejor el flujo de uso y la toma de decisiones del usuario.

Demoras en la Escritura (ya implementado): Observar el tiempo entre la escritura de caracteres individuales, como el tiempo entre escribir "H" y el resto de una palabra, para analizar la fluidez de la entrada de texto.

Para analizar la fluidez en la entrada de texto, empleamos `TactilTypingVariance`. Este método mide la variabilidad en el tiempo entre la escritura de caracteres individuales, proporcionando una perspectiva sobre la fluidez y la rapidez en la escritura.

Tiempo de Edición (ya implementado): Registrar el tiempo total dedicado a la edición de respuestas en los formularios.

Creemos que el tiempo de edición se verá reflejado en nuestras métricas de borrado y en el tiempo total registrado por widget. Estas medidas nos darán una idea del esfuerzo de edición en los formularios.

Tiempo de Scrolleo (ya implementado): Medir cuánto tiempo pasan los usuarios desplazándose a través de los formularios, lo que puede reflejar la eficiencia de la disposición del contenido.

Nuestro método `distanceMoved` acumula el tiempo que los usuarios pasan desplazándose a través de los formularios.

Cambio de Teclado (ignorado): Observar la frecuencia y el tiempo dedicado a cambiar entre diferentes teclados o modos de entrada.

Tras una cuidadosa consideración, hemos concluido que la inclusión de métricas relacionadas con el cambio de teclado en nuestra plataforma no parece ser una estrategia particularmente útil. Nuestra decisión se basa en varios factores:

En muchos casos, como en la entrada de contraseñas o datos mixtos, los cambios de teclado son una parte esperada y natural del proceso de entrada. Estos cambios no necesariamente reflejan una dificultad o retraso en la interacción del usuario. Por lo tanto, medir los cambios de teclado en estos contextos podría no proporcionar información significativa sobre la experiencia del usuario.

Los patrones de interacción de los usuarios con diferentes widgets pueden ser muy variados. Algunos usuarios pueden cambiar de teclado por preferencia personal en el caso de usar símbolos o números, más que por una necesidad impuesta por la interfaz. Esta diversidad hace que sea desafiante establecer una correlación directa y significativa entre los cambios de teclado y la eficiencia o dificultad en la interacción del usuario.

En conclusión, desde un punto de vista técnico, la implementación de un sistema para rastrear los cambios de teclado en cada widget no garantiza un retorno claro en términos de información útil o mejoras en la experiencia del usuario.

Tiempo Total de Completado (ya implementado): Cronometrar el tiempo total que toma a los usuarios completar la totalidad de cada widget.

Cada widget cuenta con su propio `focusTime`, que mide el tiempo total invertido en su completado.

En definitiva, las únicas métricas nuevas son `TextDeletionMetrics` y `TactilTouchingVariance`, de las demás consideramos que estaban cubiertas y no había necesidad de modificar nada.

4.4 Amenazas a la validez

Durante el desarrollo del capítulo 4, se presentaron algunas limitaciones y amenazas a la validez que es importante destacar. Aunque la colaboración con un reducido grupo de expertos en experiencia de usuario (UX) aportó conocimientos previos del trabajo y una comprensión profunda de los desafíos, la cantidad limitada de participantes puede haber restringido la diversidad de perspectivas.

La falta de una muestra más amplia podría haber limitado la identificación de ciertos problemas o la generalización de los resultados. Además, la participación de expertos con conocimientos previos del trabajo podría haber introducido sesgos, ya que su comprensión previa de los objetivos y métodos del estudio podría haber influido en sus evaluaciones.

Además, la interacción directa con los expertos podría haber generado un ambiente en el que ellos se sintieran inclinados a ofrecer comentarios positivos o a retener críticas constructivas, lo que podría haber sesgado la retroalimentación recibida. Este fenómeno es conocido como “sesgo de deseabilidad social”.

A pesar de estas limitaciones, la colaboración con un grupo reducido de expertos en UX permitió obtener insights valiosos y detallados que contribuyeron significativamente al

refinamiento y validación de las métricas propuestas. Sin embargo, es importante tener en cuenta estas limitaciones al interpretar los resultados y considerar cómo podrían afectar la generalización de los hallazgos en contextos más amplios.

4.5 Métricas agregadas luego del análisis de los expertos

Como se comentó anteriormente sobre los hallazgos de los expertos en UX, vamos a mostrar mas a detalle la implementación de las 2 métricas adicionales que desarrollamos a partir de sus hallazgos:

TextDeletionMetrics: Esta métrica se encarga de calcular 2 cosas: la cantidad de borrados, y la cantidad de borrados completos en los input de texto.

```
class TextDeletionMetrics extends MicroMetric {
  constructor(logger) {
    super(logger);
    this.targetElements = "input[widget-type='text']";
    this.onInput = this.onInput.bind(this);
    this.previousValues = new Map();
  }

  setUp() {
    addEventListener(this.targetElements, 'input', this.onInput);
  }
  tearDown() {
    removeEventListener(this.targetElements, 'input', this.onInput);
  }

  onInput(event) {
    const inputElement = event.target;
    const previousValue = this.previousValues.get(inputElement) || "";
    const currentValue = inputElement.value;

    if (previousValue.length > currentValue.length) {
      if (currentValue.length === 0) {
        // Borrado completo
        this.microMetricLogger.getWidgetLogs(event.target).completedDelete++;
      } else {
        // Borrado de un solo carácter
        this.microMetricLogger.getWidgetLogs(event.target).charDeleted++;
      }
    }
  }
}
```

```
this.previousValues.set(inputElement, currentValue);
}
}
```

El método `onInput` es el controlador de eventos para el evento de entrada. Se activa cada vez que se ingresa texto en un elemento de entrada correspondiente. El método realiza las siguientes acciones:

- Obtiene el elemento de entrada (`inputElement`) y los valores anterior y actual del elemento.
- Compara la longitud del valor anterior con la longitud del valor actual para determinar si se eliminó texto.
- Si la longitud anterior es mayor que la longitud actual, se considera que se eliminó texto.
- Dependiendo de la situación, actualiza las métricas de la eliminación completa de texto (`completedDelete`) o la eliminación de un solo carácter (`charDeleted`).
- Actualiza el valor anterior en el Map para su uso en la próxima llamada al método `onInput`.

TactilTouchingVariance: Esta métrica sirve para responder a la necesidad de los expertos UX sobre detectar la demora entre toques. Lo que hace la métrica es obtener intervalos de tiempo entre cada toque para luego calcular la desviación estándar con estos números, lo que nos daría al final un número que representa la variación o dispersión de los datos. Es decir, que un número grande significa que se demoró en algunas ocasiones más que el promedio de las veces.

```
touchHandler(event) {
  if (this.lastTouchTimestamp != 0) {
    var switchingTime = event.timeStamp;
    var intraKeypressInterval = switchingTime - this.lastTouchTimestamp;
    this.microMetricLogger
      .getWidgetLogs(event.currentTarget)
      .tactilIntervals.push(intraKeypressInterval);
    this.updateStandardDeviation(event);
  }
  this.lastTouchTimestamp = event.timeStamp;
}

updateStandardDeviation(event) {
  var variance = this.standardDeviation(
    this.microMetricLogger.getWidgetLogs(event.currentTarget).tactilIntervals
  );
  this.microMetricLogger.getWidgetLogs(event.currentTarget).tactilVariance =
    variance;
}
```

}

4.6 Conclusiones

En la etapa final de este estudio, se llevó a cabo un proceso meticuloso de refinamiento de las métricas propuestas, fundamentado en el análisis exhaustivo realizado por expertos en experiencia de usuario (UX). La colaboración con estos profesionales y el detallado examen de las grabaciones de los usuarios completando formularios en dispositivos móviles fueron elementos clave para identificar aspectos cruciales y perfeccionar las métricas de manera efectiva.

La creación de un espacio compartido y la organización de reuniones con los expertos permitieron una evaluación detallada y una discusión constructiva sobre el desempeño de los usuarios en los formularios. Los resultados de esta evaluación condujeron a la identificación de métricas adicionales, como el Contador de Borrados y los Registros de Misclicks, que son fundamentales para comprender la experiencia del usuario en la interacción con formularios móviles.

La incorporación de estas nuevas métricas y el refinamiento de las existentes, basados en las recomendaciones de los expertos en UX, fortalecen nuestro enfoque y mejoran la precisión de la evaluación del esfuerzo de interacción. Este proceso iterativo de revisión y mejora es esencial para desarrollar un conjunto de herramientas de evaluación que sean relevantes y efectivas en el contexto de dispositivos móviles.

Además, la implementación de métricas como TextDeletionMetrics y TactilTouchingVariance responde a las necesidades identificadas por los expertos y refleja la importancia de adaptar y refinar constantemente nuestras herramientas de evaluación para abordar los desafíos específicos del entorno móvil. Estas métricas adicionales proporcionan una mayor profundidad y precisión en la evaluación del esfuerzo de interacción, permitiendo una comprensión más matizada de la experiencia del usuario.

En conclusión, la colaboración entre investigadores y expertos en UX para refinar y validar las métricas de evaluación ha sido un componente crucial en el desarrollo de una metodología robusta para evaluar la experiencia del usuario en dispositivos móviles. Los hallazgos de esta etapa contribuyen significativamente al avance en el campo de la usabilidad móvil, ofreciendo perspectivas valiosas para optimizar el diseño de interfaces móviles y mejorar la satisfacción del usuario en la era digital.

Capítulo 5

Conclusiones

A lo largo de esta investigación, hemos abordado la necesidad de desarrollar un conjunto de métricas específicas para evaluar el esfuerzo de interacción en dispositivos móviles. Este trabajo se ha centrado en adaptar y refinar métricas existentes para el contexto móvil, así como en identificar nuevos indicadores que reflejen las particularidades de la interacción táctil.

5.1 Diferencias y Similitudes entre las Métricas Móviles y de Escritorio

Las métricas de dispositivos móviles presentan diferencias significativas en comparación con las de escritorio, principalmente debido a las características inherentes de la interacción táctil y las limitaciones de pantalla. Por ejemplo, la métrica *TactilTouchingVariance*, que mide la demora entre toques aplicando una desviación estándar como resultado, es exclusiva del entorno móvil y no tiene un equivalente directo en el escritorio. Del mismo modo, *DistanceMoved*, que calcula la distancia total recorrida con el dedo arrastrando sobre la pantalla, es una métrica específica para dispositivos táctiles.

El resto de métricas son muy similares a las de desktop, la esencia de lo que buscan calcular son lo mismo pero cambia la manera de resolverlo. Por ejemplo, la medición del tiempo de enfoque (FocusTime), la velocidad de desplazamiento (TouchVelocity), el cambio de tipo de entrada (InputSwitch), entre otras, son relevantes tanto en dispositivos móviles como en escritorio, aunque los métodos de interacción y los valores esperados puedan variar.

Durante la fase de preparación de la evaluación con participantes ([sección 3.2.1](#)), se observó que la adaptación de los formularios y la infraestructura de pruebas para dispositivos móviles fue esencial para garantizar la validez de las métricas recopiladas. Esta adaptación incluyó la implementación de un diseño responsivo y la configuración de herramientas como NGROK para facilitar el acceso remoto al backend.

5.2 Medición del Esfuerzo en Dispositivos Móviles

La evaluación del esfuerzo de interacción en dispositivos móviles ha demostrado ser factible y eficaz mediante el uso de métricas adaptadas a este contexto. La implementación de estas métricas y su validación en un entorno real han proporcionado una visión precisa del esfuerzo

de interacción y la experiencia del usuario en dispositivos móviles. Por ejemplo, la métrica `DistanceMoved`, que mide la distancia total recorrida por el dedo en la pantalla, ha sido útil para evaluar la eficiencia del desplazamiento en formularios móviles.

Una de las conclusiones principales es que, aunque el objetivo de medir el esfuerzo de interacción es común tanto en dispositivos móviles como en escritorio, las diferencias inherentes en la interacción táctil frente al uso de ratón/teclado y las dimensiones de la pantalla requieren adaptaciones específicas en las métricas. Por ejemplo, métricas como `TactilTypingVariance` y `FocusTimeTactil` son adaptaciones de sus equivalentes en escritorio, lo que demuestra que el esfuerzo se puede medir de manera similar, pero con ajustes necesarios para cada plataforma.

El contexto de uso de los dispositivos móviles, que tiende a ser más variado que el de las computadoras de escritorio, puede influir en la percepción del esfuerzo del usuario. Las pantallas más pequeñas de los dispositivos móviles afectan la forma en que los usuarios interactúan con los formularios y, por lo tanto, cómo se mide el esfuerzo. Por ejemplo, la métrica `DistanceMoved` puede tener diferentes implicaciones en móviles debido a la necesidad de desplazamientos más frecuentes.

Además, se constata que, a pesar de las diferencias en la medición del esfuerzo, la usabilidad sigue siendo un factor crítico en ambos contextos. Las métricas deben ser diseñadas cuidadosamente para capturar aspectos relevantes de la experiencia del usuario, ya sea en dispositivos móviles o en computadoras de escritorio.

En conclusión, la evaluación del esfuerzo de interacción en dispositivos móviles ha demostrado ser una tarea factible y esencial para comprender la experiencia del usuario en estos dispositivos. A través de la adaptación y validación de métricas específicas, se ha logrado una medición precisa del esfuerzo, lo que contribuye significativamente al campo de la usabilidad móvil y a la mejora de la experiencia del usuario en la era digital.

5.3 Percepción del Esfuerzo en Dispositivos Móviles

La percepción del esfuerzo en dispositivos móviles es un aspecto crucial que influye en la experiencia de usuario. Aunque existen similitudes con el entorno de escritorio, como la importancia de la facilidad de completar tareas y la fluidez de la interacción, existen factores específicos del entorno móvil que juegan un papel significativo en la percepción del esfuerzo.

Uno de estos factores es la precisión del toque. En dispositivos móviles, donde el espacio de interacción es limitado y se utiliza la interacción táctil, la precisión se vuelve crítica. La métrica `MisInteractions`, que contabiliza los toques erróneos, proporciona una visión clara de cómo la precisión de la interacción táctil afecta la percepción del esfuerzo. Es evidente que en un entorno móvil, donde el uso puede ser más incómodo que en una computadora, los usuarios son más propensos a cometer errores en la interacción.

Además, las nuevas métricas introducidas, como "TextDeletionMetrics" y "TactilTouchingVariance", han enriquecido nuestra comprensión del esfuerzo de interacción en dispositivos móviles. Estas métricas, desarrolladas a partir del feedback de expertos en UX, permiten una evaluación más detallada de la experiencia del usuario, abordando aspectos como la fluidez de la entrada de texto y la variabilidad en la interacción táctil.

En cuanto a la comparación entre la percepción del esfuerzo en dispositivos móviles y de escritorio, nuestros hallazgos sugieren que, si bien la naturaleza de las interacciones y los desafíos pueden diferir, los principios subyacentes siguen siendo similares. La facilidad de uso, la claridad de la interfaz y la fluidez de la interacción son fundamentales en ambos contextos. Sin embargo, las peculiaridades del entorno móvil, como la necesidad de una mayor precisión táctil y la adaptación a pantallas más pequeñas, requieren una consideración especial en el diseño y evaluación de la experiencia del usuario.

En resumen, la percepción del esfuerzo en dispositivos móviles se basa en factores similares a los de escritorio, pero con matices específicos del entorno móvil que deben ser abordados para optimizar la experiencia del usuario. Las métricas refinadas y validadas en este estudio proporcionan una base sólida para futuras investigaciones y desarrollos en el campo de la usabilidad móvil, con el objetivo final de mejorar la satisfacción del usuario en la era digital.

Capítulo 6

Contribuciones y Futuras Direcciones

Este estudio ha realizado contribuciones significativas al campo de la experiencia de usuario (UX) en dispositivos móviles, proporcionando un marco para evaluar el esfuerzo de interacción en este contexto. Las métricas propuestas y refinadas ofrecen herramientas valiosas para investigadores y diseñadores que buscan optimizar la experiencia del usuario en dispositivos móviles. Estas métricas permiten una comprensión más profunda de los factores que influyen en la percepción del esfuerzo y la usabilidad en entornos móviles.

Como futuras direcciones de investigación, se sugiere ampliar el alcance de las pruebas con una mayor cantidad de usuarios para recopilar más datos y mejorar la precisión de los modelos de predicción. Este enfoque permitirá una evaluación más detallada del esfuerzo de interacción y proporcionará información valiosa para el diseño de interfaces más intuitivas y eficientes.

Una vez desarrollados y validados los modelos de predicción, estos podrían emplearse directamente en la mejora de formularios y la experiencia de usuario en entornos reales. Al aplicar estos modelos, los diseñadores y desarrolladores tendrían la capacidad de identificar de manera proactiva áreas específicas de los formularios que requieren atención y optimización. Esto permitiría una iteración más rápida y efectiva en el diseño de UX, asegurando que los formularios sean intuitivos y accesibles para los usuarios finales.

Además, se plantea la posibilidad de extender la aplicación de estas métricas y modelos de predicción a un entorno más amplio, permitiendo que las pruebas se realicen desde cualquier dispositivo móvil a través de Internet o una aplicación dedicada. Esto facilitaría la recopilación de datos en un contexto más natural y diverso, mejorando aún más la relevancia y aplicabilidad de los hallazgos en el diseño de UX para dispositivos móviles.

En conclusión, este estudio abre nuevas vías para la investigación y el desarrollo en el campo de la usabilidad móvil, ofreciendo un enfoque basado en datos para la mejora de la experiencia del usuario en dispositivos móviles.

Anexo A: Detalles Técnicos y Herramientas Utilizadas

A.1 Django para el Backend

Django es un framework de desarrollo web de alto nivel escrito en Python, que promueve el desarrollo rápido y un diseño limpio y pragmático. Fue elegido para el backend de nuestra plataforma debido a sus características robustas y su capacidad para manejar aplicaciones web complejas.

Arquitectura MVT: Django sigue el patrón de arquitectura Modelo-Vista-Template (MVT), que separa la lógica de la aplicación en componentes manejables y promueve la reutilización del código.

ORM (Object-Relational Mapping): Django incluye un ORM integrado que facilita la interacción con la base de datos y la manipulación de los datos como objetos de Python, abstrayendo las complejidades de las consultas SQL.

Panel de Administración: Django proporciona un panel de administración automático y personalizable que permite a los administradores gestionar el contenido de la aplicación y las bases de datos de manera eficiente.

Seguridad: Django incluye varias características de seguridad, como la prevención de ataques de inyección SQL, cross-site scripting (XSS) y falsificación de solicitudes entre sitios (CSRF).

A.2 SQLite

SQLite es un sistema de gestión de bases de datos relacional ligero y autónomo, que se utiliza como base de datos predeterminada en proyectos Django. Fue elegido por su simplicidad y fácil integración con Django.

Sin Servidor: SQLite no requiere un servidor de base de datos separado, lo que lo hace ideal para entornos de desarrollo y pruebas.

Portabilidad: La base de datos SQLite es un archivo único, lo que facilita la copia y el traslado de la base de datos entre diferentes entornos.

Eficiencia: A pesar de su simplicidad, SQLite es capaz de manejar una cantidad significativa de consultas y operaciones de base de datos con buen rendimiento.

A.3 JavaScript y CSS para el Frontend

Para el frontend de la plataforma, se utilizó JavaScript y CSS para mejorar la interactividad y el diseño de la interfaz de usuario.

JavaScript: Se empleó para agregar dinamismo a la página web, gestionar eventos de usuario, validar formularios y realizar solicitudes asíncronas al servidor (AJAX).

CSS: Se utilizó para estilizar y organizar visualmente los elementos de la página web, mejorando la experiencia del usuario y la accesibilidad de la plataforma.

A.4 NGROK para Acceso Remoto

NGROK es una herramienta que permite exponer un servidor local a Internet, creando un túnel seguro. Se utilizó NGROK para permitir el acceso a la plataforma desde dispositivos móviles externos durante las pruebas.

Túnel Seguro: NGROK proporciona un enlace seguro y cifrado entre el servidor local y el acceso público a través de Internet.

Fácil de Usar: Con solo ejecutar un comando, NGROK genera una URL pública que se puede utilizar para acceder al servidor local desde cualquier lugar.

A.5 Scripts en Python para Exportación de Datos

Se desarrollaron scripts en Python para automatizar la exportación de datos a archivos CSV, facilitando el análisis de las métricas y la evaluación del esfuerzo del usuario.

Referencias

- Anderson C, R. , Domingos P. , Weld D, S. (2001). Personalizing web sites for mobile users. Proceedings of the 10th international conference on World Wide Web. May 2001 Pages 565–575
- A. Oulasvirta , S. De Pascale, J. Koch, T. Langerak, J. Jokinen, K. Todi, M. Laine, M. Krsthombuge, Y. Zhu, A. Miniukovich, G. Palmas, and T. Weinkauf. 2018. Aalto Interface Metrics (AIM): A Service and Codebase for Computational GUI Evaluation. In The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings. ACM, 2018: 16-19
- Badran, O. and Al-Haddad, S. (2018). The impact of software user experience on customer satisfaction. Journal of Management Information and Decision Sciences, 21(1):1–20.
- Bass ,L., John E, B. (15 June 2003) Linking usability to software architecture patterns through general scenarios. Journal of Systems and Software, Volume 66, Issue 3, 2003, Pages 187-197
- Bel B., Grill T. & Gruhn V. (22 June, 2010) Exploring the benefits of the combination of a software architecture analysis and a usability evaluation of a mobile application. Journal of Systems and Software, Volume 83, Issue 11, November 2010, Pages 2031-2044
- Bonora L. G., Grigera J., Gardey J, C. & Garrido A. (2022) Web User Interaction Speed Study. 41st International Conference of the Chilean Computer Science Society
- Cui Yanqing, Oulasvirta Antti & Ma Lingyi (2011) Event Perception in Mobile Interaction: Toward Better Navigation History Design on Mobile Devices, International Journal of Human–Computer Interaction, April 2011, Pages 413-435
- Gardey J, C. , Grigera J., Rodriguez A. , Rossi G. & Garrido A. (2022) An Interaction Effort Score for Web Pages, In Proceedings of the 18th International Conference on Web Information Systems and Technologies (WEBIST 2022), pages 439-443
- Gardey J, C. , Grigera J., Rodriguez A. , Rossi G. & Garrido A. (24 August, 2022) Predicting interaction effort in web interface widgets. International Journal of Human-Computer Studies, Volume 168, December 2022
- Grigera J., Gardey J, C. , Rodriguez A. , Rossi G. & Garrido A. (4–9 May, 2019) One Metric for All: Calculating Interaction Effort of Individual Widget, CHI'19 Extended Abstracts, Glasgow, Scotland, UK

- Hinderks, A., Winter, D., Schrepp, M., and Thomaschewski, J. (2019b). Applicability of user experience and usability questionnaires. *Journal of Universal Computer Science*, 25 (13), 1717-1735.
- Hussain, A. (2012). Metric based evaluation of mobile devices: Mobile goal question metric (mGQM). University of Salford (United Kingdom).
- Kaur, A & Kaur K. (2019). Investigation on test effort estimation of mobile applications: Systematic literature review and survey. *Information and Software Technology*, Volume 110, June 2019, Pages 56-77
- Kaur, A & Kaur K. (2022). Systematic literature review of mobile application development and testing effort estimation. *Journal of King Saud University - Computer and Information Sciences*, Volume 34, Issue 2, February 2022, Pages 1-15
- Kjeldskov J., Stage J. (May 2004) New techniques for usability evaluation of mobile systems. *International Journal of Human-Computer Studies*, Volume 60, Issues 5–6, May 2004, Pages 599-620
- Lettner, F & Holzmann C. (2011). Usability Evaluation Framework. *Computer Aided Systems Theory – EUROCAST 2011* pp 560–567
- M. Nebeling, M. Speicher, and M. Norrie. 2013. W3touch: metrics-based web page adaptation for touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2013)*. ACM, New York, NY, USA , 2311-2320.
- M. Speicher, A. Both, and M. Gaedke. 2015. S.O.S.: Does Your Search Engine Results Page (SERP) Need Help? In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing System (CHI 2015)*. ACM, New York, NY, USA, 1005-1014.
- Lars-Erik Janlert and Erik Stolterman. 2017. The Meaning of Interactivity – Some Proposals for Definitions and Measures. *Human–Computer Interaction*, 32:3, 103-138
- Sauro, J. (2015). Supr-q: A comprehensive measure of the quality of the website user experience. *Journal of usability studies*, 10(2).
- Speicher, M., Both, A., and Gaedke, M. (2014). Ensuring web interface quality through usability-based split testing. *lcwe, LNCS 8541*, pages 93–110