

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE



“RE-DISEÑO CENTRADO EN PERSONAS DE LA INTERFAZ DE DASHAI”

DANIEL ALEJANDRO ROCO VERA

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

Profesor Guía: Claudia López
Profesor Correferente: Felipe Bravo Márquez

JULIO - 2024

DEDICATORIA

A mi familia, por hacerme quien soy hoy.

AGRADECIMIENTOS

Quiero comenzar expresando mi profundo agradecimiento a mi familia, por su inquebrantable apoyo y confianza en mí a lo largo de mi trayectoria universitaria. Siempre estuvieron ahí, creyendo en mi capacidad para completar mis estudios.

Agradezco también a mis amigos, quienes fueron un pilar fundamental durante mi carrera. Su ayuda y compañía en los momentos más difíciles fueron invaluableles.

A mis respetados profesores, les debo no solo conocimientos académicos, sino también lecciones de vida que han dejado una marca indeleble en mi desarrollo profesional y personal.

No puedo olvidar mencionar a los equipos con los que tuve la oportunidad de realizar mis prácticas profesionales. Su contribución fue vital para mi crecimiento como individuo y para mejorar mis habilidades interpersonales.

Un agradecimiento especial al equipo de DashAI, quienes me acogieron con calidez y profesionalismo desde el primer día. Su actitud respetuosa y amigable hizo de mi experiencia un verdadero placer.

Mi gratitud a los participantes de las pruebas de las metodologías. Su colaboración fue fundamental para llevar a cabo el rediseño de la interfaz, y sin ellos, este logro no habría sido posible.

Por último, este trabajo fue posible gracias al apoyo en el desarrollo de DashAI de parte del Centro Nacional de Inteligencia Artificial CENIA, FB210017, BASAL, ANID.

RESUMEN

Resumen—En este trabajo se presenta el rediseño de una aplicación *no-code* para *machine learning* basado en los principios de experiencia usuaria para mejorar la interacción y el desarrollo centrado en personas. La investigación abarcó la identificación de problemas de usabilidad, la aplicación de metodologías para obtener retroalimentación de los usuarios, el diseño de prototipos de alta fidelidad y la evaluación de estos con usuarios nuevos y familiarizados. Los resultados del estudio muestran que la aplicación de técnicas de experiencia usuaria tuvo un impacto positivo y significativo en la satisfacción del usuario, además de ayudar a los desarrolladores a construir nuevas versiones de la aplicación considerando las necesidades de las personas.

Palabras Clave— interfaz, inteligencia artificial, DashAI, experiencia usuaria, metodología.

ABSTRACT

Abstract—*This work presents the redesign of a no-code application for machine learning based on user experience principles to improve interaction and person-centered development. The research involved identifying usability issues, applying methodologies to obtain user feedback, designing high-fidelity prototypes, and evaluating these with both new and familiar users. The study results show that applying user experience techniques had a positive and significant impact on user satisfaction, as well as helping developers build new versions of the application considering people's needs.*

Keywords— *interface, artificial intelligence, DashAI, user experience, methodology.*

GLOSARIO

APIs: Application Programming Interfaces.

AutoML: Automated Machine Learning.

CENIA: Centro Nacional de Inteligencia Artificial.

CNN: Convolutional Neural Network.

CSS: Cascading Style Sheets.

CSV: Comma-Separated Values.

GBM: Gradient Boosting Machine.

HTML: HyperText Markup Language.

IA: Inteligencia Artificial.

JSON: JavaScript Object Notation.

KNN: K-Nearest Neighbour.

LabUX: Laboratorio de Experiencia Usuaría.

NaN: Not a Number.

ROC: Receiver Operating Characteristic.

SUS: System Usability Scale.

SVM: Support Vector Machine.

UAI: Universidad Adolfo Ibáñez.

UC: Pontificia Universidad Católica de Chile.

UCH: Universidad de Chile.

UI: User Interface.

UTFSM: Universidad Técnica Federico Santa María.

UX: User Experience.

INDICE DE CONTENIDOS

RESUMEN.....	4
ABSTRACT	4
INDICE DE FIGURAS.....	10
INDICE DE TABLAS.....	11
INTRODUCCIÓN	12
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	13
1.1 Situación actual	13
1.2 Objetivos	15
1.2.1 Objetivo General	15
1.2.2 Objetivos Específicos	15
CAPÍTULO 2: MARCO CONCEPTUAL	16
2.1 Inteligencia artificial	16
2.1.1 Machine learning	16
2.1.2 AutoML.....	18
2.2 UX	19
2.3 Design Thinking	19
2.3.1 Contextual inquiry.....	20
2.3.2 Comparación competitiva.....	21
2.3.3 User journeys	22
2.3.4 Prototipos de alta fidelidad	23
2.3.5 Prueba de usabilidad	23
2.3.6 Pruebas exploratorias	24
2.3.7 Heurísticas de Nielsen.....	25
2.3.8 Cuestionario de usabilidad	26

2.4 Desarrollo web	27
CAPÍTULO 3: PROPUESTA DE SOLUCIÓN	29
3.1 Prueba de usabilidad para la versión vertical	31
3.1.1 Diseño	34
3.1.2 Ejecución	36
3.1.3 Recomendaciones	37
3.2 Comparación Competitiva	38
3.2.1 Diseño	39
3.2.2 Ejecución	40
3.2.3 Recomendaciones	41
3.3 Contextual Inquiry	42
3.3.1 Diseño	43
3.3.2 Ejecución	44
3.3.3 Recomendaciones	45
3.4 Pruebas Exploratorias	46
3.4.1 Diseño	46
3.4.2 Ejecución	47
3.4.3 Recomendaciones	48
3.5 Prueba de usabilidad para la versión horizontal	49
3.5.1 Diseño	51
3.5.2 Ejecución	53
3.5.3 Recomendaciones	54
3.6 Resumen de hallazgos	55
3.7 Prototipo	57
3.7.1 Implementación	59
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN	61

4.1 Prueba de usabilidad del prototipo final.....	61
4.1.1 Diseño	61
4.1.2 Ejecución	64
4.1.3 Hallazgos encontrados	65
4.1.4 Hallazgos sobre lo modificado en el prototipo	66
4.1.5 Resultados del cuestionario	66
4.2 Comparación entre la situación previa y la actual	68
CAPÍTULO 5: CONCLUSIONES	69
5.1 Trabajo a corto plazo	72
5.2 Trabajo futuro	72
5.3 Consejos para utilizar métodos UX	73
REFERENCIAS BIBLIOGRÁFICAS.....	74
ANEXOS.....	76
A.1. Invitación y Consentimiento informado	76
A.2. Prueba de Usabilidad versión vertical	78
A.3. Comentarios y observaciones	80
A.4. Respuestas de la entrevista final.....	81
B.1. User Journeys	82
B.2. Resultados de la comparación	83
C.1. Observaciones obtenidas	84
D.1. Formato utilizado	85
D.2. Problemas y mejoras encontrados	85
E.1. Prueba de Usabilidad versión horizontal.....	85
E.2. Comentarios y observaciones.....	88
E.3. Respuestas de la entrevista final	89
F.1. Recomendaciones de la carga de datos y experimentos	90

F.2. Recomendaciones de los resultados y experiencia general	90
G.1. Prueba de Usabilidad de la versión final del prototipo	91
G.2. Hallazgos	93

INDICE DE FIGURAS

Figura 1: Árbol del problema.....	14
Figura 2: <i>Design Thinking</i>	19
Figura 3: Plan de trabajo.....	29
Figura 4: Módulo de carga de datos de la versión vertical.	32
Figura 5: Módulo de experimentos de la versión vertical.....	33
Figura 6: Módulo de resultados de la versión vertical.	33
Figura 7: Página de inicio de la versión horizontal.	50
Figura 8: Módulo de carga de datos de la versión horizontal.	50
Figura 9: Módulo de experimentos de la versión horizontal.	50
Figura 10: Módulo de resultados de la versión horizontal.....	51
Figura 11: Comparación entre el módulo de resultados de la versión horizontal con la versión prototipo.	57
Figura 12: Prototipo de la elaboración de los gráficos para comparar los modelos, para cualquiera de las métricas.	58
Figura 13: Prototipo de la elaboración de los gráficos para comparar los modelos, para las métricas generales.	59
Figura 14: <i>Pull Request</i> del prototipo implementado.	60
Figura 15: Resultados del SUS.	67

INDICE DE TABLAS

Tabla 1: Tareas que el usuario realizó en la prueba de usabilidad de la versión vertical.	35
Tabla 2: Tiempo en segundos que se demoró cada usuario en completar las tareas de la prueba de usabilidad de la versión vertical.	36
Tabla 3: Mejores candidatos para cada <i>user journey</i>	41
Tabla 4: Tareas que el usuario realizó en la prueba de usabilidad de la versión horizontal.	52
Tabla 5: Tiempo en segundos que se demoró cada usuario en completar las tareas de la prueba de usabilidad de la versión horizontal.	53
Tabla 6: Tareas que el usuario realizó en la prueba de usabilidad de la versión prototipo.	63
Tabla 7: Adaptación de las preguntas SUS en DashAI.	63
Tabla 8: Tiempo en segundos que se demoró cada usuario en completar las tareas de la prueba de usabilidad de la versión final del prototipo.	64

INTRODUCCIÓN

DashAI es una aplicación en desarrollo por el CENIA, junto a profesionales y estudiantes, que busca facilitar el acceso con una interfaz gráfica interactiva a la IA de punta, tanto para analistas de datos como para estudiantes, mediante una plataforma de código abierto y extensible para el entrenamiento, evaluación, explicación y ejecución de modelos¹. Con DashAI, personas no expertas pueden implementar fácilmente modelos, técnicas y soluciones mediante AutoML.

El problema que enfrenta DashAI reside en la falta de evaluaciones con usuarios, lo que dificulta determinar si su experiencia es positiva o negativa en términos de usabilidad. Sin estas evaluaciones, es imposible saber cómo interactúan los usuarios con la aplicación, cuáles son sus dificultades y qué aspectos necesitan mejoras. Para abordar esta situación se propuso un plan de acción que consta de cuatro iteraciones.

Cada iteración se enfocó en el uso de técnicas diseñadas para obtener retroalimentación sobre la experiencia del usuario. Los resultados de estas pruebas proporcionarán recomendaciones para las futuras versiones de la aplicación y sirvieron de base para el diseño de un prototipo. Este último fue sometido a una prueba de usabilidad para determinar si los cambios realizados contribuyen a mejorar la interacción y satisfacción de los usuarios en comparación al estado inicial de la aplicación. Es importante tener en cuenta que DashAI está en constante evolución, por lo que la evaluación abarcó desde la versión inicial, denominada “versión vertical”, hasta la implementación del prototipo y su evaluación. Algunas recomendaciones se implementaron durante el transcurso del rediseño, mientras que otras quedaron en manos de los desarrolladores para su posterior desarrollo.

La estructura de este documento comienza con una contextualización y definición del problema que posee DashAI. Luego, se presentará el marco conceptual para unificar conceptos y lenguajes utilizados, seguido de los capítulos que se centran en detallar el diseño y ejecución del plan de cuatro iteraciones, así como el prototipado y su evaluación. Finalmente, se muestran las conclusiones sobre lo aprendido, el trabajo a corto plazo y consejos para quienes deseen utilizar métodos UX.

¹ Esta explicación de DashAI fue sacada de un documento para postular al concurso IDeA I+D 2023.

CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA

1.1 Situación actual

En el ámbito de la inteligencia artificial, el mercado ofrece una amplia variedad de aplicaciones diseñadas para experimentar con distintos modelos de *machine learning*. Herramientas como WEKA², RapidMiner Studio³, Perceptilabs⁴, Akkio⁵, H2O Flow⁶ y KNIME⁷ representan solo una fracción de las opciones disponibles, cada una ofreciendo sus propias capacidades únicas y especializaciones. En este entorno competitivo, es crucial para cualquier nueva herramienta que quiera entrar en la competencia, poder estar a la altura del mercado y no quedar opacado en el proceso.

Un diferenciador crítico en este escenario es la experiencia de usuario. La funcionalidad avanzada por sí sola no es suficiente, sino que la accesibilidad y el atractivo de la interfaz son igualmente importantes. Una interfaz no intuitiva o descuidada puede convertirse en un obstáculo significativo, ya que puede hacer que los usuarios no se sientan a gusto al intentar realizar sus tareas. Esto podría llevarlos a optar por otras herramientas que ofrezcan una experiencia más amigable y accesible, lo que resulta en una pérdida de usuarios potenciales y limitando el impacto en el mercado de la herramienta.

El CENIA, una corporación privada sin fines de lucro fundada por la UC, UCH, UAI y UTFSM, tiene como objetivo convertirse en un pilar para el desarrollo de la IA en Chile y Latinoamérica. Para facilitar el acceso a esta tecnología, CENIA está desarrollando una plataforma de código abierto llamada DashAI, destinada a posicionarse en el mercado actual, la cual permitirá a usuarios sin conocimientos avanzados en programación desarrollar soluciones con *machine learning*. Sin embargo, enfrenta desafíos críticos relacionados con su interfaz y experiencia de usuario. Diseñada principalmente por expertos en IA, la plataforma no ha sido probada exhaustivamente con usuarios finales, lo que genera incertidumbre sobre su recepción por parte del público objetivo. Este enfoque podría resultar en una experiencia desfavorable para usuarios menos experimentados, afectando negativamente la adopción y éxito de la herramienta en el mercado.

Además, el desarrollo continuo de la aplicación introduce una complejidad adicional. Aunque los cambios en la interfaz entre versiones no han sido drásticos, la constante incorporación de nuevas funcionalidades crea incertidumbre sobre cuál será la versión definitiva presentada al público. Esta situación resalta la importancia de definir claramente las características esenciales que deben estar presentes en la versión final para satisfacer las expectativas de los usuarios. Es crucial determinar cuáles funcionalidades deben ser priorizadas y cuáles pueden ser abordadas en futuras

² <https://ml.cms.waikato.ac.nz/weka/>

³ <https://altair.com/altair-rapidminer>

⁴ <https://www.perceptilabs.com/>

⁵ <https://www.akkio.com/>

⁶ <https://docs.h2o.ai/h2o/latest-stable/h2o-docs/flow.html>

⁷ <https://www.knime.com/>

actualizaciones, asegurando que las necesidades inmediatas de los usuarios no se vean comprometidas.

Para superar estos desafíos de manera efectiva, es crucial adoptar un enfoque centrado en el usuario durante todo el proceso de diseño. Esto implica realizar pruebas exhaustivas con usuarios reales y utilizar diversos métodos para identificar y priorizar las funcionalidades que satisfacen sus necesidades. Basar las decisiones de diseño en una comprensión profunda de la experiencia usuario asegura una interfaz funcional, eficiente, intuitiva y atractiva. Este enfoque mejorará la satisfacción y retención de usuarios, y reducirá la incertidumbre sobre la versión final, garantizando que las nuevas características sean bien recibidas. En última instancia, esto posicionará mejor a DashAI en el mercado, aumentando su competitividad y adopción.

Luego de identificar los desafíos a los que se enfrenta esta aplicación, especialmente con su interfaz y el impacto en la experiencia del usuario, se procederá a presentar en la Figura 1 un “Árbol del Problema”. Este diagrama ayuda a comprender de manera más clara el problema, ilustrando las causas y efectos que limitan el éxito de esta aplicación en el mercado.

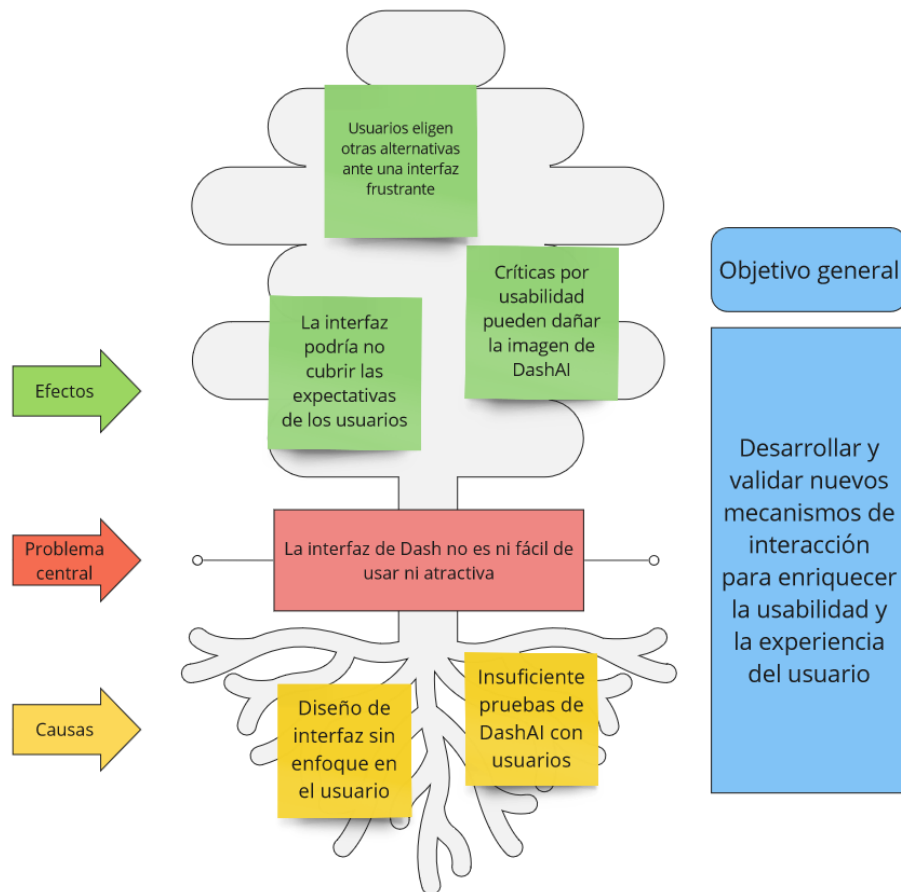


Figura 1: Árbol del problema.

Fuente: Elaboración propia.

1.2 Objetivos

Para el desarrollo y mejora de DashAI, se establecen los siguientes objetivos:

1.2.1 Objetivo General

Desarrollar y validar nuevos mecanismos de interacción para enriquecer la usabilidad y la experiencia usuaria de la aplicación DashAI.

1.2.2 Objetivos Específicos

- Analizar la interacción de usuarios no familiarizados con el sistema DashAI a través de pruebas de usabilidad y entrevistas diseñadas para ello.
- Proponer un rediseño para uno de los módulos de DashAI a través de mecanismos de interacción, acordes a la evidencia obtenida en el análisis anterior.
- Evaluar la experiencia usuaria con el prototipo propuesto, principalmente mediante pruebas de usabilidad.

CAPÍTULO 2: MARCO CONCEPTUAL

El marco conceptual que se presentará a continuación estará enfocado en proporcionar una guía completa sobre los conceptos, herramientas y metodologías necesarias para prototipar y desarrollar una aplicación de inteligencia artificial efectiva y atractiva para el usuario.

2.1 Inteligencia artificial

La IA es un campo de la informática que se enfoca en el desarrollo de sistemas y programas que pueden realizar tareas que, de otra manera, requerirían de la inteligencia humana para llevarlas a cabo [Rouhiainen18].

En el ámbito académico y empresarial, se utiliza en una amplia gama de aplicaciones, que van desde la automatización de tareas rutinarias [Kumar12] hasta la creación de sistemas de diagnóstico médico avanzados [Litjens17], la optimización de procesos industriales [Plathottam23], el desarrollo de vehículos autónomos [Grigorescu19] y mucho más. Su objetivo es mejorar la eficiencia, la precisión y la capacidad de adaptación de los sistemas tecnológicos en diversos ámbitos.

En esencia, busca replicar y automatizar procesos cognitivos y de toma de decisiones que normalmente asociamos con la inteligencia humana. Utiliza una variedad de técnicas, como el *machine learning*, el procesamiento del lenguaje natural, la visión por computadora y la planificación, entre otras, para lograr este objetivo.

2.1.1 Machine learning

Es una rama de la inteligencia artificial que se enfoca en el desarrollo de algoritmos y modelos que permiten a las computadoras aprender patrones y realizar tareas específicas a partir de datos, sin necesidad de ser programadas explícitamente para cada tarea.

La idea fundamental es que los sistemas informáticos puedan mejorar automáticamente su rendimiento en una tarea específica a medida que se les proporciona más información. Esto se logra mediante el análisis de grandes conjuntos de datos, en los cuales el algoritmo identifica patrones y relaciones estadísticas que pueden ser utilizadas para realizar predicciones o tomar decisiones.

Existen diversas implementaciones, ante lo cual se detallará un flujo que explique el paso a paso de cada sección utilizada.

Este proceso inicia con la carga de los datos desde diversas fuentes, tales como archivos de texto, bases de datos y APIs. Posteriormente, se realiza una exhaustiva verificación de la calidad de estos, corrigiendo valores nulos o inconsistentes, eliminando filas con información incompleta y llevando a cabo otras operaciones necesarias para garantizar la integridad de la información.

Una vez completada la fase anterior, los datos se dividen en tres conjuntos: entrenamiento, validación y prueba. El conjunto de entrenamiento se utiliza para entrenar el modelo, permitiéndole aprender patrones y relaciones dentro de los datos. Por último, el conjunto de prueba se reserva para evaluar el rendimiento final del modelo una vez que ha sido entrenado y ajustado.

La siguiente etapa corresponde a seleccionar un algoritmo de *machine learning* adecuado para el problema en cuestión, considerando factores como la naturaleza de los datos y los objetivos del proyecto. Algunos algoritmos comunes incluyen SVM, CNN y GBM. Una vez seleccionado, se definen los hiperparámetros a utilizar, que son configuraciones que afectan su comportamiento y desempeño.

Tras efectuar los ajustes iniciales, se inicia el entrenamiento del modelo utilizando el conjunto respectivo. Durante este proceso, se monitorea continuamente su desempeño para evitar el sobreajuste, es decir, asegurando que no aprenda en exceso detalles específicos de los datos, lo que podría limitar su capacidad de generalización.

Después del entrenamiento inicial, se realizan modificaciones iterativas para mejorar el rendimiento en el conjunto de validación. Se prueban distintas configuraciones y se evalúa su impacto.

Una vez logrado, se procede a evaluar el modelo con el conjunto de prueba. Se calculan diversas métricas de evaluación, como precisión, sensibilidad y puntuación F1, para proporcionar una medida cuantitativa del rendimiento en la tarea específica.

Finalmente, luego de que ha sido preparado, ajustado y validado adecuadamente, está en condiciones de ser desplegado para su uso práctico. Las predicciones que produce pueden ser empleadas directamente en situaciones reales, lo que permite tomar decisiones fundamentadas basadas en la información disponible.

Existen diversas categorías en el campo del *machine learning* [Gerón19] que se clasifican según:

- Entrenamiento con o sin supervisión.
- Capacidad para aprender gradualmente o necesitar todo el conjunto de datos.
- Si compara datos nuevos con los conocidos o detecta patrones para predecir sin comparar directamente.

Dentro de la categoría de aprendizaje con supervisión humana, se encuentra el aprendizaje supervisado, donde el algoritmo se entrena con datos etiquetados por humanos, aprendiendo a realizar predicciones o clasificaciones a partir de ejemplos. Es relevante destacar que DashAI, principalmente, utiliza este tipo de aprendizaje en su funcionamiento.

2.1.2 AutoML

Tal como se presentó anteriormente, las fases por las que hay que pasar para generar predicciones a través de un modelo de *machine learning* pueden resultar complicadas para usuarios inexpertos o personas no familiarizadas con el tema. Es por ello por lo que existen herramientas como DashAI, que automatizan gran parte de este proceso, permitiendo a usuarios sin experiencia en programación o ciencia de datos construir modelos complejos de manera más rápida y sencilla. Automatizan tareas como la selección de algoritmos, la optimización de hiperparámetros y la evaluación del modelo, lo que les permite enfocarse en interpretar los resultados y tomar decisiones basadas en ellos.

Las opiniones respecto al uso de herramientas automatizadas para generar predicciones o sus componentes pueden variar entre las personas, ya sea por razones éticas o profesionales. Esto suscita interrogantes sobre qué aspectos deberían ser automatizados por una aplicación.

Para lo anterior es que se evalúa el trabajo de Dakuo Wang [Wang21], que a través de una encuesta online utilizando un *framework* AutoML logran argumentar que el nivel de automatización y el tipo de explicación de un sistema puede variar mucho dependiendo de quién es el usuario, en qué etapa de su ciclo de vida trabajo y cuál es su tarea. No hay necesidad de desarrollar trabajos de ciencia de datos completamente automatizados, es más, se agradece la posibilidad de que algunos elementos no sean dirigidos solamente por la máquina, sino también por el usuario, lo cual le permite tener más confianza en la herramienta, pero no se descarta la posibilidad en un futuro de variar tal nivel de manipulación.

Otro escrito corresponde a "*Human-AI collaboration in data science*" [Wang19]. La creencia general del grupo entrevistado es que siempre tendrá que haber algún nivel de retroalimentación de una mente humana para un sistema de aprendizaje automatizado, porque simplemente no puede inferir la pericia en la materia pertinente para comprender el significado de datos de un conjunto de datos. Además, poseer un sistema completamente automático llama la atención de los gerentes, debido a los ahorros de costos asociados con la necesidad de contratar menos trabajadores, pero dudan de que alguna vez pueda replicar a un profesional en el área.

Basándose en los hallazgos de Dakuo Wang y su equipo, así como en la reflexión sobre la colaboración entre humanos y sistemas automatizados en "*Human-AI collaboration in data science*", DashAI podría considerar un enfoque flexible en su rediseño de interfaz. Esto implica ofrecer opciones tanto para la automatización como para la interacción humana en diferentes etapas del proceso de ciencia de datos. Permitir al usuario tener cierto control sobre el proceso y su visualización podría aumentar la confianza en la herramienta, mientras que la integración de explicaciones claras y comprensibles ayudaría a respaldar la toma de decisiones informadas [Drozdal20].

2.2 UX

La experiencia usuaria abarca todas las sensaciones y emociones al interactuar con una empresa, producto o servicio, trascendiendo la simple usabilidad. Su objetivo es optimizar esta interacción, facilitando la búsqueda de información, ejecución eficiente de tareas y proporcionando una experiencia fluida y placentera, a través de un enfoque iterativo que incluye investigación, creación de prototipos y pruebas constantes. Un buen diseño UX se centra en satisfacer las necesidades humanas básicas, tales como la autonomía, competencia, estimulación (centrada en uno mismo), relación y popularidad (centrada en los demás), durante la interacción con el producto o servicio [Hassenzahl08].

Para la ejecución efectiva de proyectos que demanden un flujo constante de información y adaptaciones, se requiere la implementación de una metodología que permita captar datos en diferentes momentos de su ciclo de vida, por lo que a continuación se mostrará una que contribuye a esto.

2.3 Design Thinking

El *Design Thinking* es un proceso iterativo y no lineal que implica una retroalimentación constante en cada etapa, tal cómo se puede ver en la Figura 2. Esta metodología se enfoca en comprender al usuario, sus desafíos y necesidades, y luego ajustar progresivamente las soluciones para cumplir con sus expectativas [Lewrick20]. Es esencial mantener una comunicación continua con las partes interesadas en todas las etapas del proceso.

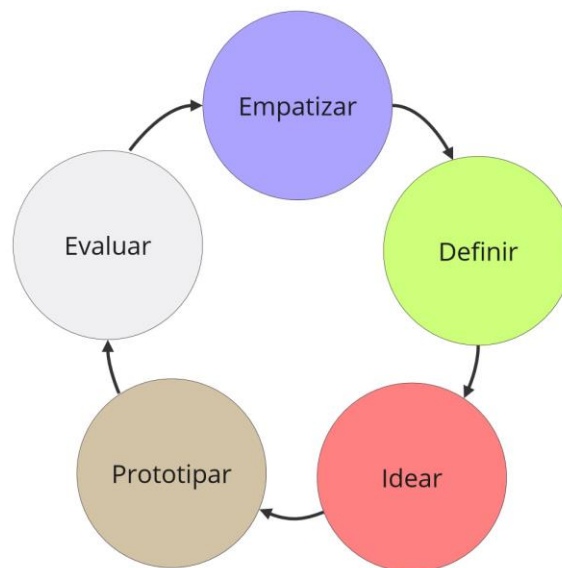


Figura 2: *Design Thinking*.

Fuente: Elaboración propia.

Aunque la implementación de esta metodología puede variar según el público objetivo y los recursos disponibles, generalmente sigue cinco pasos iterativos:

1. Empatizar: Comprender el problema mediante la investigación y la escucha activa, centrándose en las necesidades del público objetivo y fomentando su participación en el proceso creativo.
2. Definir: Focalizar el problema identificado en la fase de empatía, estableciendo un objetivo claro y creando perfiles para representar a los beneficiarios de la solución, si es necesario.
3. Idear: Generar múltiples soluciones creativas para abordar el problema, utilizando técnicas como la lluvia de ideas y anotando todas las propuestas posibles.
4. Prototipar: Construir prototipos para dar forma a las ideas, experimentando y refinando sin buscar validación definitiva, con el objetivo de identificar opciones y soluciones viables.
5. Evaluar: Probar las soluciones propuestas en contextos relevantes, obteniendo retroalimentación de equipo y usuarios a través de entrevistas, pruebas y dinámicas colaborativas, para iterar y mejorar el diseño si es necesario.

DashAI puede aprovechar esta estrategia como una metodología integral que abarque diversas técnicas de UX, permitiéndole obtener una valiosa retroalimentación y diseñar soluciones centradas en el usuario de manera efectiva.

En las siguientes subsecciones se mostrarán los distintos métodos a utilizar, los cuales apoyan cada uno de los pasos del *Design Thinking*, partiendo por un método que apoya el paso de empatizar.

2.3.1 Contextual inquiry

Es una técnica cualitativa que se enfoca en la observación detallada y las entrevistas colaborativas para entender las interacciones de los usuarios con sistemas y productos en su entorno natural [Salazar20]. A través de este método, se busca descubrir información oculta sobre el trabajo de los clientes, que puede no ser accesible mediante otros enfoques de investigación.

Se distingue por su atención al contexto en el que ocurren las interacciones y por la búsqueda activa de comprensión sobre cómo y por qué los usuarios realizan ciertas acciones. Su objetivo principal es obtener una visión profunda y precisa de las prácticas laborales y los comportamientos de los usuarios para informar el diseño de soluciones centradas en el usuario.

Los pasos por seguir para realizar una investigación contextual pueden variar según el propósito definido, sin embargo, se puede adoptar un enfoque específico como el propuesto por Basusta [Basusta15]:

1. Identificar preguntas clave de investigación: Definir preguntas relacionadas con la tarea general o el flujo de trabajo que se esté investigando, manteniéndolas amplias para abordar los objetivos y resultados esperados.

2. Encontrar participantes: Buscar a personas que estén familiarizadas o sean expertas en la herramienta o plataforma que se está evaluando.
3. Desempeñar un papel: Adoptar el rol de “maestro y aprendiz”, donde los usuarios serán los maestros y el investigador será el aprendiz. Es importante realizar preguntas de qué hacen, cómo lo hacen y por qué lo hacen.
4. Identificar atajos o heurísticas utilizadas por los participantes: Observar y tomar nota de cualquier acción interesante, inesperada o confusa que realice el participante.
5. Revisar las notas y reflexionar: Analizar las notas para comprender la función de cada tarea y buscar formas de simplificarlas o incluso eliminarlas en la medida de lo posible.

En cuanto al paso de definir, se muestra a continuación un método utilizado para focalizar la aplicación enfrente de otras similares.

2.3.2 Comparación competitiva

En el capítulo anterior sobre la definición del problema, se explicó que DashAI se enfrenta al desafío de competir en un mercado donde ya existen herramientas consolidadas. Por lo tanto, es crucial entender qué elementos ofrecen estas herramientas y cómo esta aplicación puede aprovecharlos para posicionarse efectivamente. Para abordar este desafío, es necesario desarrollar una estrategia adecuada.

La Comparación Competitiva, también conocida como análisis competitivo, es un método de investigación estratégica que se utiliza para evaluar y comprender la posición relativa de un producto, servicio o marca en relación con sus competidores dentro y fuera de la industria.

Para establecer un proceso efectivo, se puede tomar como referencia a Hooley [Hooley20], quien tiene en cuenta los siguientes aspectos a la hora de realizar el análisis:

- Identificar con quién compararse: Seleccionar los competidores que se evaluarán. Esta selección se realiza con el propósito de analizar diferentes enfoques y descubrir áreas de mejora y oportunidades de innovación.
- Identificar qué aspectos del negocio comparar: Se deben priorizar aquellos procesos clave a lo largo de la cadena de valor que impacten significativamente en la satisfacción del cliente y en los costos operativos, y que presenten un alto potencial de mejora.
- Recopilar datos relevantes: Recolectar información del propio producto y el de los competidores. Esto puede darse a través de fuentes publicadas, intercambio de información o mediante el uso del producto o servicio.

- Comparar con los procesos propios: Se comparan los procesos de los competidores con el producto propio, identificando acciones y procedimientos que podrían mejorarse o adoptarse para mejorar la competitividad.

Si bien inicialmente esta técnica puede no parecer relacionada directamente con UX, es importante destacar que mediante un proceso estructurado es posible diseñar usuarios ficticios para simular la ejecución de tareas específicas. Estas simulaciones permiten evaluar aspectos clave desde la perspectiva del usuario, considerando sus necesidades como el criterio central.

En la fase de idear del *Design Thinking*, es esencial transformar la comprensión del usuario en soluciones concretas. A continuación, se presenta un método que permite visualizar y analizar el recorrido del usuario, ayudando a generar ideas creativas y mejorar la experiencia usuaria.

2.3.3 User journeys

También conocidos como viajes del usuario, son representaciones detalladas y sistemáticas del proceso que un usuario sigue al interactuar con un producto, servicio o sistema digital. Estos viajes describen cada paso realizado, desde el momento en que se encuentra por primera vez con el producto hasta que completa su objetivo o tarea específica, incluyendo sus reacciones y emociones.

Su objetivo principal es proporcionar una comprensión detallada de cómo los usuarios interactúan con una empresa o producto en múltiples puntos de contacto, lo que permite identificar áreas de mejora, puntos de fricción y oportunidades para optimizar la experiencia [Kaplan23].

Para ilustrar más fácilmente el recorrido es posible utilizar *user journey mapping* [Endmann16], que consiste en crear una representación visual o narrativa del viaje del usuario, que ilustra el proceso con:

- Su desencadenante y objetivo.
- Todas las actividades por ejecutar.
- Todas las personas involucradas.
- Requisitos conocidos, casos de uso, problemas, entre otros.

Hay que considerar que se pueden basar en usuarios reales o ficticios, dependiendo de la disponibilidad de datos y de los objetivos del análisis. Además, aunque los *user journeys* suelen incluir emociones experimentadas por los usuarios en cada etapa, es posible que en los casos de usuarios ficticios estas sean estimadas en base a conocimientos previos sobre el público objetivo y no reflejen necesariamente emociones reales.

El prototipado es la etapa en la que se generan diversas opciones y soluciones viables que deben discutirse con el equipo para definir una versión final. En este contexto, se presenta un método comúnmente utilizado junto con un ejemplo de herramienta.

2.3.4 Prototipos de alta fidelidad

El prototipado de alta fidelidad es esencial en el proceso de diseño al crear representaciones detalladas y precisas de la interfaz de usuario final. Estos prototipos simulan de manera realista la experiencia del usuario interactuando con el producto digital. Mediante herramientas avanzadas de diseño gráfico y prototipado interactivo como Figma⁸, los equipos validan la usabilidad, estética y funcionalidad del diseño antes de su implementación. Esto implica desarrollar elementos de interfaz coherentes e integrar interacciones dinámicas entre pantallas para asegurar que el diseño cumpla con los requisitos del usuario y las expectativas del cliente. A través de iteraciones y refinamientos, se busca optimizar la experiencia del usuario y reducir riesgos en el desarrollo del producto.

A continuación, se mostrarán varios métodos para evaluar las soluciones generadas y el estado del producto.

2.3.5 Prueba de usabilidad

Es un método para evaluar la facilidad de uso de un producto o sistema observando cómo las personas interactúan con él [Lewis12].

En una Prueba de Usabilidad, los usuarios son invitados a completar tareas específicas mientras interactúan con el producto o sistema. Durante esta interacción, se recopila retroalimentación a través de observaciones directas, comentarios verbales y respuestas a preguntas formuladas durante la sesión. El objetivo principal es identificar problemas y áreas de mejora en el servicio. Al comprender directamente cómo interactúan con el producto, los diseñadores pueden detectar obstáculos o confusiones que podrían existir en la interfaz, como dificultades para encontrar funciones, comprender las instrucciones o completar tareas de manera eficiente.

Dos técnicas comunes utilizadas en las Pruebas de Usabilidad son *think aloud* y *verbal probing*. Con la primera, los usuarios expresan en voz alta sus pensamientos mientras interactúan con el producto, lo que proporciona información valiosa sobre cómo perciben y comprenden la interfaz. Por otro lado, la segunda implica hacer preguntas específicas a los usuarios sobre su experiencia y decisiones durante la prueba, lo que ayuda a los diseñadores a profundizar en sus procesos mentales y detectar áreas problemáticas con mayor precisión [Geisen17].

Para realizar pruebas efectivas, es crucial seguir un proceso estructurado que permita obtener resultados significativos y aplicables. De acuerdo con la metodología que utiliza Barnum en su libro [Barnum11], se han identificado una serie de pasos clave que abarcan desde la planificación hasta la presentación de los hallazgos.

⁸ <https://www.figma.com/>

1. **Planificación:** Durante esta fase, se establecen los objetivos y métodos de prueba, se eligen los participantes y se fijan las fechas. Además, se redacta el plan de prueba, que servirá como guía para la ejecución del proceso de usabilidad.
2. **Preparación:** Aquí se concentran los esfuerzos en reclutar a los participantes adecuados, asignar roles y preparar los materiales necesarios para la prueba. Se elaboran guiones y cuestionarios que guiarán la interacción durante la sesión.
3. **Conducción:** En esta etapa se lleva a cabo la prueba propiamente dicha. Se configura el entorno, se dirige la sesión y se registran todas las observaciones relevantes. La moderación es clave para garantizar la efectividad y objetividad del proceso.
4. **Análisis:** Una vez recopilados los datos, se procede a analizarlos en busca de patrones y tendencias. Se identifican los hallazgos más significativos y se formulan recomendaciones basadas en estos resultados.
5. **Informe:** Finalmente, se redacta un informe detallado que documenta los hallazgos de manera clara y efectiva. Este informe se presenta a las partes interesadas, junto con las recomendaciones correspondientes, con el objetivo de mejorar la usabilidad del producto.

Otro método para evaluar las soluciones corresponde a las pruebas exploratorias.

2.3.6 Pruebas exploratorias

Son un método de evaluación que busca comprender cómo los usuarios interactúan con un producto o sistema digital de forma más libre y sin restricciones predefinidas. En lugar de seguir un guion o un conjunto específico de tareas, los usuarios tienen la libertad de explorar el producto de la manera que prefieran.

Entre sus objetivos se encuentran:

- **Identificar problemas de usabilidad:** Permiten descubrir problemas que podrían dificultar el uso del producto, como errores de navegación, elementos de la interfaz confusos o funciones poco intuitivas.
- **Comprender las necesidades y expectativas de los usuarios:** Al observar a los usuarios interactuar con el producto, se puede obtener información valiosa sobre sus necesidades, expectativas y comportamientos.
- **Generar ideas para mejorar el diseño:** Los comentarios de los usuarios pueden servir como inspiración para nuevas ideas y soluciones de diseño que mejoren la experiencia general del usuario.
- **Evaluar la viabilidad de un concepto de diseño:** Las pruebas exploratorias pueden ayudar a determinar si un concepto de diseño es viable y si tiene el potencial de ser exitoso con los usuarios objetivo.

Algunas de sus ventajas son:

- Flexibilidad: Al no estar atados a un guion, los investigadores pueden observar comportamientos inesperados y descubrir problemas que podrían pasarse por alto en pruebas más estructuradas [Sachedina17].
- Eficiencia: Las pruebas exploratorias pueden ser más eficientes que las pruebas estructuradas, ya que no requieren tanto tiempo para preparar y ejecutar.
- Profundidad: Permiten una exploración más profunda de la usabilidad del producto, ya que los usuarios tienen la libertad de seguir sus propios caminos y descubrir áreas que los investigadores podrían no haber considerado.

Por otro lado, también tienen algunas desventajas que hay que considerar:

- Falta de estructura: La naturaleza abierta de las pruebas exploratorias puede dificultar la comparación de resultados entre diferentes estudios o metodologías.
- Sesgo del moderador: El moderador de la prueba puede influir en el comportamiento de los usuarios, por lo que es importante que esté bien entrenado y sea imparcial.
- Dificultad de moderación: Moderar una prueba exploratoria puede ser un desafío, ya que requiere que el moderador siga el ritmo de los usuarios y esté atento a los posibles problemas de usabilidad.

También existe otro método, el cual corresponde a una evaluación experta, es decir, que implica la revisión crítica del diseño por parte de especialistas en usabilidad y experiencia del usuario.

2.3.7 Heurísticas de Nielsen

Desarrolladas por Jakob Nielsen, un experto danés en usabilidad web y diseño de interfaces de usuario y cofundador de Nielsen Norman Group, una consultora líder en investigación de usabilidad, experiencia de usuario y diseño de interacción. Estas heurísticas representan un conjunto de principios fundamentales para evaluar la usabilidad de las interfaces de usuario. Al aplicarlas, los diseñadores y evaluadores pueden identificar áreas de mejora y tomar decisiones informadas para optimizar la usabilidad y la experiencia del usuario.

A continuación, se detallan las diez heurísticas de Nielsen:

- Visibilidad del Estado del Sistema: Mantener a los usuarios informados sobre qué está pasando a través de retroalimentación adecuada dentro de un tiempo razonable.
- Coincidencia entre el Sistema y el Mundo Real: Hablar el lenguaje de los usuarios, utilizando términos familiares en lugar de jerga de sistema.

- Control y Libertad para el Usuario: Permitir a los usuarios deshacer acciones no deseadas y navegar fácilmente por el sistema.
- Consistencia y Estándares: Mantener la consistencia en todo el sistema, siguiendo convenciones de la plataforma y los estándares de la industria.
- Prevención de Errores: Diseñar el sistema de manera que se eviten errores en primer lugar, mediante mensajes de error claros y acciones reversibles.
- Reconocimiento más que Recuerdo: Minimizar la carga de memoria del usuario, haciendo que las opciones, acciones y objetos sean visibles y fáciles de encontrar.
- Flexibilidad y Eficiencia de Uso: Ofrecer atajos para usuarios experimentados, así como soporte para usuarios novatos, sin complicar la interfaz.
- Diseño Estético y Minimalista: Buscar la simplicidad y la concisión en el diseño, evitando la información irrelevante o redundante.
- Ayudar a los Usuarios a Reconocer, Diagnosticar y Recuperarse de Errores: Proporcionar mensajes de error claros y soluciones para ayudar a los usuarios a resolver problemas.
- Ayuda y Documentación: Ofrecer información clara y concisa para ayudar a los usuarios a comprender el sistema y realizar tareas de manera efectiva.

Una de las ventajas de utilizarlas es que proporcionan criterios específicos para evaluar la usabilidad de un sistema de manera más ordenada. Esto puede ser útil para organizar los resultados de experimentos que carecen de una estructura definida y proporcionar retroalimentación adecuada a los diseñadores y otras partes interesadas.

Por último, la recolección de la opinión del usuario sobre la experiencia con un producto digital mediante preguntas específicas es fundamental para evaluar la usabilidad y mejorar la interacción.

2.3.8 Cuestionario de usabilidad

Es un método utilizado por organizaciones, empresas o instituciones para recopilar información sobre la opinión, percepción y nivel de satisfacción de los clientes, empleados u otros grupos de interés. Estas encuestas suelen incluir preguntas relacionadas con la calidad de productos o servicios, la atención al cliente, la eficacia de procesos, entre otros aspectos relevantes para la experiencia del encuestado.

El propósito principal es obtener una retroalimentación directa que permita a la organización evaluar su desempeño, identificar áreas de mejora y comprender las necesidades y expectativas de sus públicos.

Existen cuestionarios ya diseñados, tales como el SUS para evaluar la usabilidad de un *software*, sitio web o aplicación. Este consta de diez preguntas que los usuarios deberán

completar después de interactuar con el producto, usando para ello la escala de Likert ("Totalmente en desacuerdo" hasta "Totalmente de acuerdo"):

- 1) Creo que me gustaría utilizar este sistema con frecuencia.
- 2) Encontré el sistema innecesariamente complejo.
- 3) Pensé que el sistema era fácil de usar.
- 4) Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema.
- 5) Encontré que las diversas funciones de este sistema estaban bien integradas.
- 6) Pensé que había demasiada inconsistencia en este sistema.
- 7) Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente.
- 8) Encontré el sistema muy complicado de usar.
- 9) Me sentí muy seguro usando el sistema.
- 10) Necesitaba aprender muchas cosas antes de empezar con este sistema.

Cada pregunta tiene una escala de 1 a 5 en la escala de Likert, lo cual se utilizará para realizar el cálculo de la puntuación SUS, cuya fórmula es:

- 1) Se debe sumar las respuestas de los enunciados impares y después restar 5.
- 2) Se debe sumar las respuestas de los enunciados pares y restar ese total a 25.
- 3) Sumar ambos resultados y multiplicarlo por 2,5.

A partir de las respuestas, se obtiene una puntuación que varía entre 0 y 100. Una puntuación alta indica que los usuarios encuentran el producto fácil de usar, intuitivo y satisfactorio. En contraste, una puntuación baja sugiere que el producto es difícil de usar, frustrante y poco intuitivo. El puntaje promedio ronda los 68 puntos y un puntaje superior o igual a 80 indica una usabilidad excelente.

Para que los usuarios puedan interactuar con la aplicación, es necesario desarrollar mediante código la interfaz y funcionalidades detectadas en los métodos anteriores, todo esto considerando un ambiente de desarrollo web.

2.4 Desarrollo web

Abarca todas las etapas necesarias para crear, diseñar y mantener sitios o aplicaciones. Esto implica desde la fase inicial de planificación del diseño, la creación de la interfaz y la estructuración de la información, hasta la implementación de código, programación y optimización.

En el caso específico del rediseño de la interfaz de DashAI, se enfocará en la parte del *frontend*, que es la parte visible para los usuarios y con la que interactúan mediante su respectiva UI. Para este propósito, se utilizarán las siguientes tecnologías:

- HTML⁹: Es el lenguaje estándar utilizado para crear la estructura y el contenido básico de una página web. Se utiliza para definir los elementos y la organización de la información en una página web, como títulos, párrafos, enlaces, imágenes, formularios, entre otros.
- CSS¹⁰: Es un lenguaje de estilo que se utiliza para controlar la presentación y el diseño visual de una página web. Permite definir el aspecto y el formato de los elementos HTML, como el color, la fuente, el tamaño, el espaciado, la disposición y otros aspectos relacionados con la apariencia de la página.
- JavaScript¹¹: Es un lenguaje de programación de alto nivel que se utiliza principalmente para agregar interactividad y funcionalidad dinámica a las páginas web. Se utiliza para manipular el contenido HTML y CSS, responder a eventos del usuario, realizar validaciones de formularios, realizar animaciones, cargar contenido de forma dinámica, entre otras funcionalidades.

Además de estas tecnologías, para el desarrollo se emplearán:

- React.js¹²: Es una biblioteca de JavaScript desarrollada por Facebook que se utiliza para construir interfaces de usuario interactivas y dinámicas. Su enfoque basado en componentes permite dividir la interfaz en elementos reutilizables, lo que facilita la creación y mantenimiento de aplicaciones web complejas.
- Plotly¹³: Es una biblioteca de visualización de datos interactiva que ofrece una amplia gama de gráficos interactivos y herramientas de visualización de datos. Se utiliza para crear visualizaciones de datos dinámicas y personalizables que respondan a la interacción del usuario.

⁹ Más información sobre HTML y su función en la estructura de páginas web: <https://www.w3schools.com/html/>

¹⁰ Detalle adicional sobre CSS y su papel en el diseño visual de páginas web: <https://www.w3schools.com/css/>

¹¹ Explicación ampliada sobre JavaScript y sus capacidades en la interactividad web: <https://www.w3schools.com/js/>

¹² Descripción de React.js y su utilidad en la construcción de interfaces de usuario: <https://es.react.dev/>

¹³ Información adicional sobre Plotly y sus características como biblioteca de visualización de datos: <https://plotly.com/>

CAPÍTULO 3: PROPUESTA DE SOLUCIÓN

En este capítulo se expondrá el procedimiento que se siguió para la generación y posterior implementación de un prototipo basado en evidencia con métodos centrados en usuarios. Para alcanzar este objetivo, se estableció un plan de acción que divide los métodos y herramientas a emplear en cada fase del ciclo de vida del proyecto. En la Figura 3 es posible observar el plan que fue utilizado, el cual consta de cuatro iteraciones.

Durante la primera iteración el enfoque fue recabar, mediante una prueba de usabilidad, *feedback* de la experiencia usuaria para la interfaz inicial, la cual se caracteriza por seguir un flujo vertical en el paso a paso y que corresponde con la que ya estaba implementada antes de iniciar el proyecto de rediseño. Este método es crucial para comprobar si lo desarrollado hasta ese momento es de utilidad para usuarios con diferentes niveles de conocimientos en *machine learning*, así como para conocer sobre nuevas funcionalidades que podrían ser incorporadas a posterior.

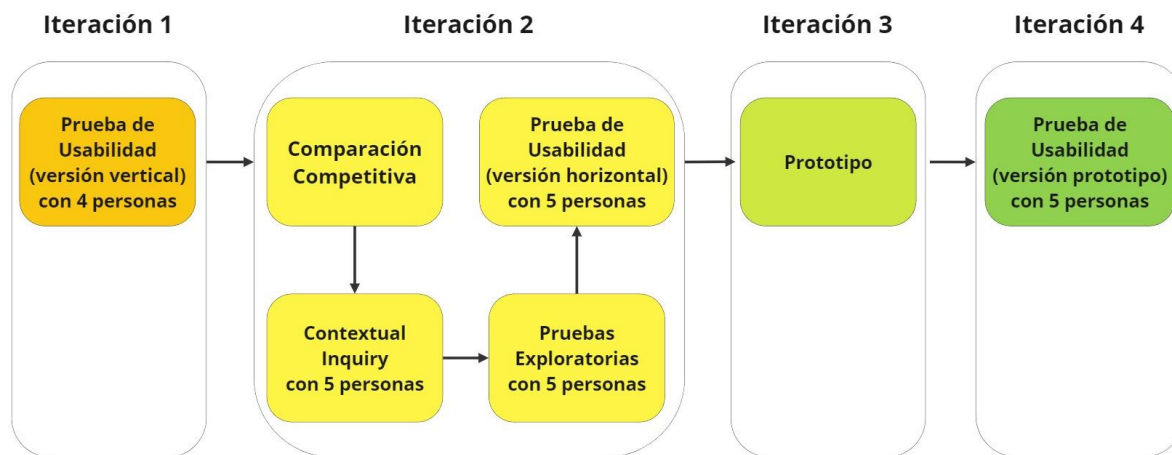


Figura 3: Plan de trabajo.

Fuente: Elaboración propia.

En la segunda iteración, se llevó a cabo cuatro métodos distintos, cada uno con un fin en particular.

- La comparación competitiva fue utilizada para entender cómo la competencia maneja funcionalidades clave, estableciendo un marco para asegurar que DashAI alcance o supere estándares de mercado.
- El *contextual inquiry* permitió investigar cómo los usuarios que trabajan con *machine learning* generan modelos sin AutoML, proporcionando *insights* valiosos sobre las expectativas y necesidades naturales de los usuarios.

- Las pruebas exploratorias y la prueba de usabilidad adicional evaluaron la interfaz de la versión horizontal. La primera detectó errores y mejoras potenciales con usuarios poco experimentados entorno a IA, mientras que la segunda evaluó las funcionalidades incorporadas en fases anteriores.

La tercera iteración resumió todos los hallazgos encontrados en los pasos anteriores y se seleccionó un módulo de la interfaz para aplicar un rediseño, generando un prototipo con Figma que posteriormente se introdujo en la aplicación mediante código. Cabe destacar que, a la par de esta implementación, los desarrolladores del proyecto DashAI estuvieron trabajando en aplicar las mejoras propuestas sobre las otras secciones de la aplicación que no fueron seleccionadas para esta memoria.

En la cuarta iteración se llevó a cabo la validación del prototipo implementado, junto con todas las instancias de desarrollo creadas hasta el momento de la ejecución, utilizando para ello una prueba de usabilidad que involucró a usuarios que hayan participado en alguno de los métodos anteriores a esta etapa. La finalidad será identificar cualquier problema de usabilidad restante y confirmar que los nuevos mecanismos de interacción son intuitivos y eficaces.

La decisión de dividir el plan de acción en cuatro iteraciones se basó en la necesidad de cumplir con los objetivos del rediseño. La primera y segunda iteración se centró en analizar la interacción de usuarios no familiarizados con DashAI y recopilar información útil para sugerir mejoras. En la tercera fase, se utilizaron los hallazgos anteriores para rediseñar la interfaz, basándose en los mecanismos de interacción previamente evaluados. Finalmente, la cuarta iteración se dedicó a evaluar la experiencia del usuario con el prototipo, validando así los nuevos mecanismos implementados. Este enfoque integral contribuyó al objetivo general de mejorar la usabilidad, la experiencia del usuario y las funcionalidades de la aplicación. Además, cabe mencionar que el prototipado y su implementación requirió una iteración específica, debido a la necesidad de múltiples reuniones con el equipo de DashAI para su finalización.

La estructura del proyecto de rediseño de DashAI sigue de cerca los principios del *Design Thinking*. Cada etapa, desde la investigación inicial hasta la implementación y validación del prototipo, se enfoca en comprender en profundidad las necesidades de los usuarios, generar ideas innovadoras y validarlas mediante prototipos iterativos. Esta metodología se seleccionó deliberadamente como el marco estructural para guiar y optimizar los métodos empleados en el proyecto, asegurando así un enfoque coherente y efectivo para mejorar la experiencia del usuario y la funcionalidad de la aplicación.

La propuesta no incluyó cambios en los algoritmos, la lógica de procesamiento, ni el rendimiento de DashAI. El enfoque se basó en mejorar la usabilidad y accesibilidad, asegurando una interfaz funcional y atractiva para los usuarios finales.

Para asegurar una comunicación fluida, se establecieron reuniones diarias y semanales. Las reuniones diarias, breves y dirigidas al equipo de desarrollo, sirvieron para revisar los avances del día anterior y establecer los objetivos diarios. Las reuniones semanales, con

participación del equipo de desarrollo y otros miembros de DashAI, permitieron discutir en detalle los progresos, desafíos y estrategias a largo plazo del proyecto.

El material generado durante el proceso, incluidos los prototipos, resultados de pruebas y recomendaciones, fueron registrados y compartidos en el Notion y Discord creados para el proyecto. Estas plataformas sirvieron como repositorios centrales para documentar y organizar el trabajo realizado, garantizando así la accesibilidad y la colaboración eficiente entre los miembros del equipo. Del mismo modo, se utilizó Github para almacenar y administrar el código asociado a la aplicación.

Con el objetivo de proporcionar recomendaciones pertinentes para futuras iteraciones de DashAI, se ha determinado que los principales módulos de la aplicación, en los que se centraron todas las actividades del plan, fueron:

- Carga de datos: Centrado en cargar un archivo y seleccionar los *split*.
- Experimentos: Sección en la cual se crean, configuran y ejecutan distintos experimentos. La diferencia entre experimentos y modelos radica en que los primeros están enfocados en resolver una *task* específica (como análisis tabular, procesamiento de texto, análisis de imágenes, entre otros), mientras que los modelos son los algoritmos utilizados dentro del experimento para resolver dicha tarea seleccionada.
- Resultados: Se muestran los valores obtenidos en las distintas métricas para cada experimento.

De igual manera se consideró el ámbito general de la experiencia del usuario en la aplicación. Se aclara que durante la primera iteración se trabajó con una versión anterior de la interfaz, por lo que las recomendaciones fueron ajustadas adecuadamente para las etapas posteriores.

En los subcapítulos siguientes se profundizará en los métodos llevados a cabo en cada iteración, dejando la prueba de usabilidad del prototipo construido para la validación de la propuesta de rediseño.

3.1 Prueba de usabilidad para la versión vertical

La primera prueba de usabilidad se llevó a cabo de manera presencial, mediante una reunión entre un evaluador y un usuario. Durante esta, se realizó una serie de actividades diseñadas para evaluar el desempeño de la interfaz y sus funcionalidades actuales.

Es importante tener en consideración las limitaciones de esta versión:

- Solo acepta archivos con extensión *.json* como *input*
- Solo permite cargar un archivo a la vez
- Solo se puede ejecutar un experimento a la vez

- Solo muestra los resultados del primer modelo ejecutado

Para ilustrar la interfaz de la aplicación se presentará un ejemplo resumido de ejecución, el cual no contempla la predicción del modelo con valores ingresados directamente por el usuario.

Primero, se debe cargar un archivo en la aplicación, la cual automáticamente determina el tipo de *task* a resolver, que en este ejemplo es una clasificación numérica, tal como se muestra en la Figura 4. Una vez cargado el archivo, se debe presionar el botón “Next” para avanzar al siguiente módulo.

A continuación, se selecciona un modelo del menú “*model type*” para resolver la tarea de clasificación numérica. Luego, se agrega el modelo, el cual aparecerá en la tabla inferior de experimentos. Cada modelo tiene parámetros predeterminados, pero es posible configurarlos pulsando el botón de configuración correspondiente. Una vez seleccionado el modelo y ajustados los parámetros, el usuario debe presionar el botón “Run Experiment” para proceder al siguiente módulo, tal como se observa en la Figura 5.

Finalmente, en pantalla aparecen los resultados del entrenamiento y la prueba tras ejecutar el experimento, junto con los parámetros seleccionados en el paso anterior. La Figura 6 muestra un ejemplo de estos resultados.

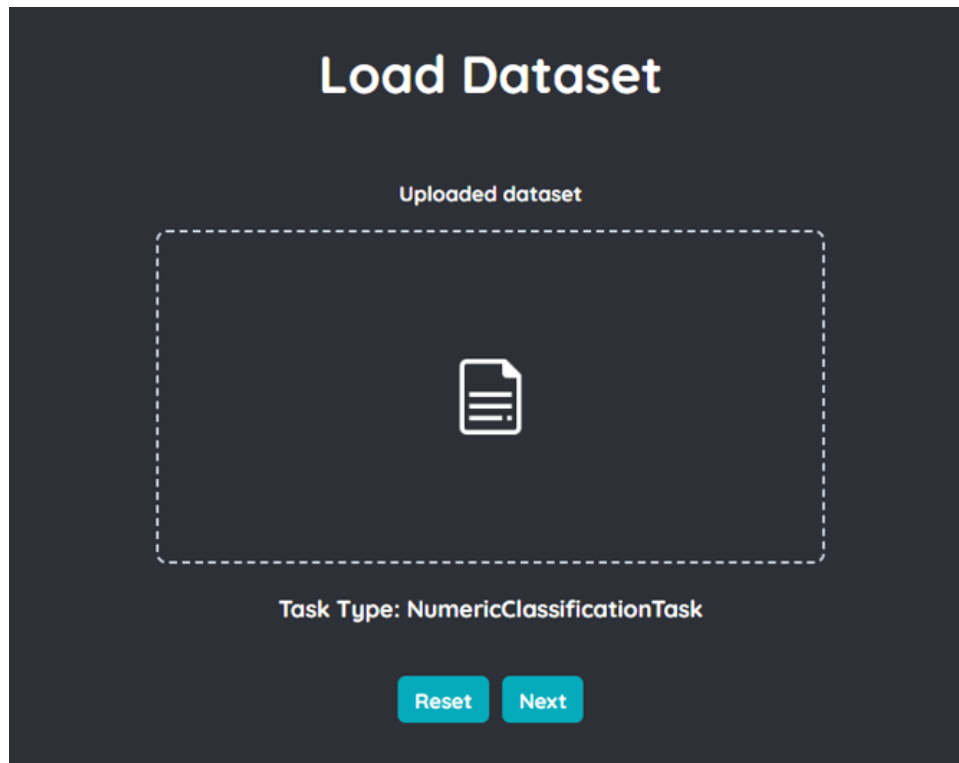


Figura 4: Módulo de carga de datos de la versión vertical.

Fuente: Aplicación de DashAI.

Add Models

nickname (optional)

model type
KNN

Add model

#	Name	Type	Parameters
0		KNN	Configure

Run Experiment

Figura 5: Módulo de experimentos de la versión vertical.

Fuente: Aplicación de DashAI.

Model: knn

Train Results	96.67%
Test Results	96.67%

Parameters

- algorithm: auto
- leaf_size: 30
- metric: minkowski
- metric_params: null
- n_jobs: null
- n_neighbors: 5
- p: 2
- weights: uniform

Next

Figura 6: Módulo de resultados de la versión vertical.

Fuente: Aplicación de DashAI.

El flujo descrito anteriormente, fue utilizado con los usuarios, en donde detectaron distintos problemas en la realización de las tareas. Además, surgieron varios comentarios sobre la interfaz y posibles mejoras sobre los elementos actuales y también con los que desearían tener en futuras versiones. El diseño utilizado para realizar la entrevista a los usuarios se verá en la siguiente subsección.

3.1.1 Diseño

Este método fue aplicado a cuatro personas con diferentes niveles de conocimiento en torno al *machine learning*. Tres de ellos son estudiantes de pregrado de la UTFSM en la sede de Valparaíso, y el cuarto es un exalumno y actual estudiante de postgrado de la misma universidad. Entre los participantes, dos tienen menos de un año de experiencia en el área, uno tiene dos años de experiencia y el último tiene más de tres años de experiencia.

Las pruebas se llevaron a cabo en el LabUX, utilizando la sala de *testing* donde se reunió únicamente el entrevistador y el usuario. Estas reuniones fueron monitoreadas y grabadas desde otra habitación con la finalidad de tomar notas de la sesión y su posterior evaluación. Además, en la primera de estas pruebas de usabilidad, un grupo de alumnos de una profesora actuaron como observadores en la sala de monitoreo para proporcionar retroalimentación adicional, no sobre la interfaz, sino sobre la calidad y contenido de la prueba, con el objetivo de mejorar la calidad de las futuras pruebas. Cada usuario recibió un consentimiento informado donde se notificó la grabación de la sesión, únicamente para tomar notas y detalles de esta. Se les dio la opción de abandonar la entrevista en cualquier momento o de solicitar la eliminación de fragmentos o totalidad de lo grabado. El documento utilizado para esto se puede encontrar en el Anexo A.1.

El flujo de la reunión consistió primero en un saludo, seguido de una introducción a DashAI y el propósito de la prueba. Luego se realizó una entrevista inicial para recabar información sobre la experiencia en el área, seguida por la ejecución de distintas tareas de usuario y, finalmente, una entrevista final sobre la experiencia en la interfaz.

Centrándose en el contenido de la reunión, se evaluaron los aspectos de la interfaz para la versión vertical de la aplicación mediante cuatro tareas desarrolladas. Estas se enfocaron en las funcionalidades de la carga de datos, la ejecución de un experimento, la observación de los resultados de un experimento y la predicción con valores nuevos sobre un modelo. El documento utilizado de guía por el entrevistador se puede encontrar en el Anexo A.2.

Las tareas específicas que los usuarios desarrollaron se detallan en la Tabla 1.

Tabla 1: Tareas que el usuario realizó en la prueba de usabilidad de la versión vertical.

Fuente: Elaboración propia.

ID	Tarea	Escenario/Instrucciones	Criterio de Término	Requerimientos
T1	Subir un archivo al sistema.	Se proporciona al usuario la interfaz de la aplicación ya lista para ejecutarse. Una vez dentro, debe ir a la sección indicada por el supervisor y proceder a cargar el archivo.	Aparece el documento cargado junto con los botones <i>Reset/Next</i> .	Archivo: irisDataset.json
T2	Predecir con valores predeterminados.	Una vez realizada la tarea anterior el usuario debe elegir el modelo a utilizar y posteriormente ejecutar el experimento.	Aparece en pantalla los resultados y los parámetros.	
T3	Insertar nuevos valores.	Finalizada la predicción, debe ir a la parte final de la interfaz e insertar un nuevo valor entregado por el supervisor.	Se muestra la predicción realizada.	Insertar los valores: 2, 2, 1, 5
T4	Predecir con valores modificados.	El usuario debe devolverse al inicio de la interfaz, cargando nuevamente el archivo de la primera tarea. Ahora debe elegir un modelo y cambiar su configuración según estime conveniente.	Aparecen en pantalla los resultados y parámetros.	Archivo: irisDataset.json

Una vez finalizadas todas las pruebas con los usuarios, se procedió a realizar un resumen de cada una de las reuniones, en el cual se registraron todas las recomendaciones y comentarios identificados durante el proceso. Estas observaciones fueron recopiladas y consolidadas en un informe detallado, presentándose a continuación.

3.1.2 Ejecución

Se registró el tiempo que cada usuario tardó en completar cada tarea, como se detalla en la Tabla 2. El Usuario 1 logró completar las tareas T2 y T3 en menos tiempo en comparación con los demás participantes. Esto se debió a que él mismo buscó y visualizó el archivo que necesitaba subir, lo que agilizó el proceso posterior.

Tabla 2: Tiempo en segundos que se demoró cada usuario en completar las tareas de la prueba de usabilidad de la versión vertical.

Fuente: Elaboración propia.

Usuario/Tarea	T1	T2	T3	T4
Usuario 1	79	15	14	127
Usuario 2	63	46	160	181
Usuario 3	32	41	98	112
Usuario 4	68	20	130	102

Durante la ejecución de cada tarea, se registraron los comentarios directos de los participantes junto con las observaciones del entrevistador. De estos registros, surgieron una serie de patrones importantes que se mencionan a continuación:

- Algunos usuarios necesitaron consultar el archivo a utilizar para saber con qué tipos de datos se enfrentarían, lo cual interrumpió su flujo de trabajo.
- En la tarea relacionada con la inserción de nuevos datos en un modelo ya entrenado, la mayoría de los usuarios mostraron incomodidad al tener que consultar el archivo para conocer el formato de los datos a ingresar, aun cuando habían consultado con anterioridad el archivo en la primera tarea.
- Se presentaron algunos errores en la aplicación que no eran notificados, lo que dejó a algunos usuarios sin saber cómo resolverlos, aumentando su desconcierto y provocando en algunos casos el reinicio de la aplicación.

Los comentarios detallados recopilados se pueden encontrar en el Anexo A.3.

Por último, durante la entrevista final se recopiló información sobre diversas características de la aplicación mediante una serie de preguntas. A continuación, se resumen las respuestas de los usuarios:

- 1) La mayoría prefiere una disposición horizontal de la interfaz por sobre la vertical.
- 2) Todos los usuarios estuvieron de acuerdo en que los elementos visuales eran adecuados.

- 3) Los usuarios sugirieron mostrar los datos entrenados hasta cierto momento o realizar un entrenamiento en paralelo si es que existiera una opción para detener un modelo.
- 4) Los usuarios desean una explicación de la utilidad de cada parámetro en el modelo.
- 5) Los usuarios expresaron interés en incluir gráficos tras finalizar el entrenamiento.

Los números anteriores corresponden a las preguntas realizadas, las cuales se detallan en el Anexo A.4. En este anexo, se pueden encontrar las nueve preguntas formuladas, organizadas de izquierda a derecha según su número.

3.1.3 Recomendaciones

Luego de analizar la información obtenida, se generaron las siguientes recomendaciones para los módulos de la carga de datos, experimentos y resultados, considerando también la experiencia general (facilidad de uso, claridad de la interfaz, satisfacción, entre otros) del usuario al utilizar la aplicación a lo largo de las tareas.

En la carga de datos

- Mostrar los primeros datos del *dataset* y columnas.
- Permitir centrar los datos entre valores los valores -1 y 1 (si es que aplica).
- Permitir el uso de OneHotEncoder, normalizadores y otras técnicas para escalar datos.

En experimentos

- Incluir un botón para detener un modelo en ejecución.
- Incluir una barra de progreso durante la ejecución.
- Incluir una descripción de la funcionalidad de cada modelo.
- Mostrar los resultados parciales del entrenamiento y prueba de un experimento tras cancelarlo.
- Incluir una opción de configuración avanzada sobre los parámetros de los modelos, que permitan al usuario modificar parámetros más específicos que la configuración básica actual.
- Incluir una opción para entrenar modelos mientras se terminan de ejecutar otros.
- Mostrar los límites entre los cuales se puede modificar cada parámetro.
- Indicar en qué afectan los parámetros al modelo si es que se modifican.

En resultados

- Incluir gráficos para visualizar el rendimiento de un modelo.
- Incluir el número de elementos utilizados en cada *split*.
- Visualizar cuáles fueron los datos erróneos en la validación.
- Visualizar únicamente los parámetros utilizados para un modelo y no todos los predeterminados.
- Diferenciar visualmente los parámetros por defecto y los modificados por el usuario.
- Almacenar el modelo en Pickle o JSON.

En la experiencia general al utilizar la aplicación

- Incluir una descripción de qué es lo que significa cada parámetro.
- Añadir un enlace a la documentación de Sklearn.
- Mostrar los valores con un máximo de dos decimales.
- Modificar los botones de incrementar y reducir valores para que sean más grandes.
- Validar los errores antes de continuar hacia otro componente.
- Enumerar las secciones de la interfaz y mostrar siempre en dónde está el usuario.

Esta prueba identificó tanto puntos fuertes como áreas de mejora en la funcionalidad y usabilidad de la aplicación en su primera versión, destacando la necesidad de un rediseño estratégico. No solo es crucial recibir retroalimentación directa de los usuarios al utilizar la aplicación, sino también evaluar cómo las aplicaciones competidoras manejan las funcionalidades clave de DashAI para adoptar las mejores prácticas identificadas.

3.2 Comparación Competitiva

Para comparar DashAI y los distintos competidores se llevó a cabo un análisis cualitativo de cómo cumplen con los distintos *user journeys* que se definieron para su evaluación, los cuales representan funcionalidades y características que DashAI posee. Cada uno de estos estableció la ruta para evaluar cada interfaz, indicando si una característica está presente en la interfaz, si está parcialmente presente o simplemente no existe dentro de esa aplicación. Se debe tener en cuenta que la evaluación de estos competidores se realizó de dos formas. La primera consistió en instalar o acceder directamente la aplicación y comprobar la ejecución de las actividades planteadas, mientras que la segunda forma implicó la visualización de videos explicativos sobre cómo se desarrolla. Esta última forma se aplicó en todas las evaluaciones, pero en caso de que la aplicación fuera de pago o si su instalación resulta demasiado complicada, se tomó como la única forma de evaluar.

Para poder realizar esta comparación, se necesitaron identificar algunos de los competidores de DashAI y establecer claramente cuales funcionalidades son las evaluadas.

3.2.1 Diseño

Para esta comparación, se han considerado los siguientes competidores en base a varias categorías clave:

Herramientas de gestión del ciclo de vida del *machine learning*, enfocadas en el seguimiento, reproducción y colaboración de experimentos:

- MLFlow¹⁴
- Weights & Biases¹⁵

Plataformas de código abierto para *machine learning* y la minería de datos, que ofrecen una amplia gama de herramientas para el análisis y creación de modelos:

- WEKA¹⁶
- RapidMinerStudio (*Educational*)¹⁷
- KNIME¹⁸

Plataformas basadas en la nube que ofrecen interfaces sin código o de bajo código, lo que las hace accesibles a usuarios sin experiencia en programación:

- Akkio¹⁹
- H2O Flow²⁰
- HuggingFace Auto Train²¹
- Módulos AI²²
- Perceptilabs²³

La comparación se llevó a cabo en función de ocho *user journeys* creados por el evaluador, cada uno de los cuales se caracteriza por un nombre descriptivo, los cuales fueron:

¹⁴ <https://mlflow.org/>

¹⁵ <https://wandb.ai/site>

¹⁶ <https://ml.cms.waikato.ac.nz/weka/>

¹⁷ <https://altair.com/altair-rapidminer>

¹⁸ <https://www.knime.com/>

¹⁹ <https://www.akkio.com/>

²⁰ <https://docs.h2o.ai/h2o/latest-stable/h2o-docs/flow.html>

²¹ <https://huggingface.co/autotrain>

²² <https://www.modulos.ai/>

²³ <https://www.perceptilabs.com/>

- 1) Cargar un *dataset* tabular
- 2) Ejecutar un experimento
- 3) Configurar un experimento
- 4) Visualizar los resultados de un experimento
- 5) Predecir con un modelo
- 6) Usar los resultados de experimentos anteriores
- 7) Cerrar la interfaz mientras se entrenan modelos
- 8) Reabrir la interfaz luego de entrenar modelos

Los pasos necesarios para la ejecución de cada uno se muestran de manera detallada en el Anexo B.1.

Las características no abordadas por los *user journeys* se resumieron en un apartado sobre la interfaz, al igual que la complejidad de la instalación (si es que aplicó) también fue analizada. Luego de diseñar la comparación competitiva, se ejecutó, tal como se observa en la siguiente subsección.

3.2.2 Ejecución

La comparación fue realizada por un evaluador que utilizó cada una de las aplicaciones seleccionadas, realizando todos los pasos de cada *user journey* propuesto. Para cada recorrido, se documentaron las observaciones durante y después de la ejecución, sumado a si se cumplieron, no se cumplieron o no se pudo medir la ejecución de los pasos.

Algunos patrones interesantes fueron:

- La mayoría de las aplicaciones carecía de una funcionalidad de predicción sobre un modelo generado, a diferencia de DashAI, que sí lo poseía.
- La mayoría de las aplicaciones incluían gráficos, mientras que DashAI no los tenía.
- La mayoría permitía guardar los archivos subidos para su uso posterior, algo que DashAI recién comenzó a implementar en su versión horizontal.

El detalle de la evaluación sobre todas las aplicaciones se puede encontrar en el Anexo B.2. En este anexo se puede encontrar una presentación visual, que brinda una visión clara y concisa de cómo cada aplicación se desempeñó en relación con las actividades medidas, permitiendo una rápida identificación de sus puntos fuertes y áreas de mejora.

Para cada uno de los *user journeys* se decidió cual aplicación era la mejor candidata, para así comprender mejor las prácticas más efectivas implementadas por estas. Para ello, se seleccionaron todas aquellas que cumplieron con todos los pasos de sus respectivos *user journeys*, además de no haber tenido mayores dificultades en su ejecución. Luego de esto,

se presentaron ante el equipo de desarrollo en una reunión semanal, en la cual se votó democráticamente cual sería considerada la mejor. En la Tabla 3 se pueden ver los seleccionados por cada uno de los *user journeys*.

Tabla 3: Mejores candidatos para cada *user journey*.

Fuente: Elaboración propia.

User journey	Mejor candidato
Cargar un <i>dataset</i> tabular	Perceptilabs
Ejecutar un experimento	HuggingFace
Configurar un experimento	H2O Flow
Visualizar los resultados de un experimento	Weights & Biases
Predecir con un modelo	H2O Flow
Usar los resultados de experimentos anteriores	MLFlow
Cerrar la interfaz mientras se entrenan modelos	HuggingFace
Reabrir la interfaz luego de entrenar modelos	HuggingFace

3.2.3 Recomendaciones

Luego de analizar la información obtenida en la comparación competitiva, y de los competidores mejor evaluados para cada uno de los *user journeys*, se han generado las siguientes recomendaciones para los módulos de la carga de datos, experimentos y resultados, considerando también la experiencia general (facilidad de uso, claridad de la interfaz, satisfacción, entre otros) del usuario al utilizar la aplicación a lo largo de las tareas.

En la carga de datos

- Añadir una visualización general de los datos cargados.
- Añadir un botón para iniciar un experimento rápidamente tras cargar los datos.
- Permitir cargar datos desde internet (Kaggle, HF, etc).
- Permitir ignorar la primera fila de los datos (en caso de que se utilice .csv)
- Incluir una semilla de ejecución para poder replicar los resultados.
- Visualizar los primeros cinco a diez datos al cargar el archivo.

- Permitir filtrar algunas filas de los datos cargados.
- Indicar visualmente la correcta carga de los datos.
- Permitir variar la distribución de los *splits* en un experimento.
- Indicar el formato que deben tener los datos para poder cargarlos.
- Permitir crear un tag a un *dataset*.
- Agregar colores diferentes para cada tipo de tarea.

En experimentos

- Mostrar cuáles son los modelos más populares de acuerdo con el tipo de tarea.
- Agregar una barra de progreso del experimento.
- Indicar a qué columna se está haciendo la predicción (variable objetivo).
- Permitir al usuario elegir la variable objetivo.
- Filtrar filas y columnas para un modelo en específico antes de ejecutar un experimento (para el caso de archivos .csv).
- Permitir crear un tag a un experimento para poder diferenciarlo.

En resultados

- Indicar la cantidad de elementos que fueron correctamente clasificados.
- Añadir gráficos para comparar distintos modelos.

En la experiencia general al utilizar la aplicación

- Incluir un tutorial que le muestre al usuario el flujo a seguir para utilizar la aplicación.

No solo es crucial entender cómo la competencia opera, sino también conocer las expectativas y necesidades naturales de los usuarios al trabajar con *machine learning* sin herramientas adicionales, es decir, utilizando la programación directa. Para esto, se empleó un método centrado en captar las necesidades del usuario en su entorno natural.

3.3 Contextual Inquiry

La realización de un *contextual inquiry* para entender las interacciones de los usuarios con las herramientas de inteligencia artificial en su entorno natural es una estrategia valiosa para identificar secciones, parámetros y elementos que podrían ser incorporados o modificados en el rediseño. Sin embargo, es importante reconocer que presentó limitaciones durante su ejecución.

La limitación principal surgió de la naturaleza de las tareas establecidas para el desarrollo durante este método. En este caso, las actividades estuvieron guiadas por un profesor hacia un grupo de estudiantes, lo que pudo restringir la espontaneidad y libertad de los participantes para expresar sus necesidades y comportamientos auténticos. Esta dirección predeterminada pudo influir en las respuestas de los entrevistados, potencialmente sesgando los datos recopilados y afectando la validez de los resultados obtenidos.

Además, la falta de libertad para explorar diferentes opciones y realizar acciones no prescritas pudo haber limitado la capacidad de los investigadores para descubrir necesidades subyacentes y problemas no detectados que podrían ser esenciales para DashAI.

Por lo tanto, es crucial abordar estas limitaciones de manera proactiva durante el proceso de *contextual inquiry*. Se debió tener cuidado al interpretar los resultados obtenidos, considerando el contexto específico de las tareas dirigidas y el grado de libertad de los participantes. Además, se deben buscar oportunidades para recopilar información adicional a través de técnicas complementarias para obtener una comprensión más completa y equilibrada de las necesidades y preferencias de los usuarios.

Es importante destacar lo que no fue estudiado. No se abordaron las circunstancias académicas individuales de cada estudiante, como su currículo específico, experiencia previa o calidad de la enseñanza recibida. Tampoco se exploraron las motivaciones personales de los estudiantes para realizar la tarea ni se investigó cómo colaboran o trabajan en grupo para resolverla. Además, no se llevaron a cabo comparaciones cuantitativas estrictas de rendimiento entre los estudiantes, ya que el enfoque principal radicó en comprender el proceso y las decisiones tomadas, no en clasificar su productividad.

3.3.1 Diseño

Los participantes seleccionados para el estudio fueron estudiantes pertenecientes a la UTFSM, quienes hayan sido parte de alguna asignatura relacionada con inteligencia artificial y demuestren un cierto nivel de competencia para abordar las tareas asignadas por sus respectivos profesores, quienes contaban con una experiencia entre cero y más de tres años.

Cada estudiante tuvo la tarea de resolver un problema relacionado a redes neuronales, proporcionado por su respectivo profesor, utilizando principalmente la biblioteca Sklearn de Python. Estuvieron divididos en dos grupos de dos personas y un estudiante trabajando de manera individual. Aunque las tareas asignadas por el docente difirieron en los resultados esperados para cada equipo, el contenido y las áreas de evaluación fueron consistentes.

La reunión presencial con estos grupos se llevó a cabo una sola vez, pero los participantes informaron sobre su progreso y dificultades al finalizar la tarea. Organizar reuniones regulares fue complicado debido a que tienen un plazo de entrega de varias semanas, y los estudiantes podrían reunirse en cualquier momento para avanzar, incluso durante los

fin de semana. Por lo tanto, se enfatizó en estudiar el inicio de la tarea, que incluyó la carga inicial de los datos para realizar los ejercicios propuestos.

Para esta investigación, se realizó una observación no participativa por parte del investigador, limitada a tomar nota de los comentarios entre los participantes (en el caso de las parejas) y de la interacción ocasional del participante con el investigador. Las reuniones no fueron grabadas para respetar la privacidad y promover un entorno cómodo para la discusión, ya que la grabación podría limitar o influir en sus comentarios. Los resultados obtenidos serán resumidos a continuación.

3.3.2 Ejecución

La sesión de reunión con los estudiantes abordó las actividades del procedimiento de carga y exploración de datos, selección del *train/test split*, valores faltantes e imputación y la estandarización. La creación de la red neuronal y su posterior modificación fueron temas que los estudiantes comentaron en días posteriores a la reunión, ya que en una sola sesión no es factible abordar por completo la totalidad de la tarea.

El resumen de los procedimientos observados en cada actividad fue:

Carga y exploración de datos

La atención de los participantes se centró en asegurarse de que los datos sean cargados correctamente, revisando si había errores durante la importación y verificando que todas las columnas y filas estuviesen presentes. Además, visualizaron la estructura de los datos, tipos de columnas, y la presencia de valores nulos o atípicos.

Selección del *train / test split*

La mayoría siguió la proporción sugerida de 70% para entrenamiento, 20% para validación y 10% para prueba. Después de realizar la división, los estudiantes verificaron el tamaño de cada conjunto para confirmar que la separación se realizó correctamente.

Valores faltantes e imputación

Inicialmente los estudiantes identificaron la existencia de valores faltantes y los reemplazaron con NaN. Posteriormente, realizaron un conteo de los valores faltantes en cada columna y visualizaron estos conteos mediante un gráfico de barras.

Estandarización

Los participantes crearon nuevos conjuntos de datos estandarizados para el entrenamiento y validación.

Creación de la red neuronal

La mayoría de los estudiantes construyeron modelos utilizando una arquitectura de red neuronal con al menos una capa oculta. La elección de la cantidad de neuronas y la función de activación varió, pero hubo una tendencia a utilizar la activación ReLu para

estas capas. Además, algunos alumnos analizaron el rendimiento de su modelo, identificando posibles casos de *underfitting* u *overfitting*, y ajustando parámetros como la tasa de aprendizaje para mejorar los resultados.

Modificación de la red neuronal

Algunos estudiantes probaron diferentes inicializadores de pesos para observar sus efectos en el aprendizaje y la estabilidad de la red. Además, implementaron otras técnicas para reducir el *overfitting*, tales como la regularización de los datos de entrada y la manipulación de la arquitectura de la red.

En el Anexo C.1 se presentan todas las observaciones obtenidas en sus respectivas actividades.

3.3.3 Recomendaciones

Los resultados obtenidos se centraron en las redes neuronales, un modelo que no está previsto implementar a corto plazo en la aplicación de DashAI, pero que podría ser considerado en el futuro. Esto permitió obtener una visión de lo que los usuarios necesitarían para llevarlo a cabo. Además, varios de los resultados podrían ser aplicables a otros modelos, ya que no se centran únicamente en su creación, sino en los ajustes necesarios de los datos para iniciar el proceso de modelado.

Una vez recopiladas todas las observaciones del *contextual inquiry* y sus consideraciones, se procedió a transformar los resultados de las actividades en elementos que podrían ser relevantes para el rediseño, enfocados en los módulos de carga de datos, experimentos y resultados, los cuales se presentan a continuación:

En la carga de datos

- Incluir una visualización previa de los datos cargados.
- Incluir un resumen estadístico del *dataset*.
- Incluir una alerta sobre valores atípicos o desequilibrados.
- Incluir gráficos sobre los datos cargados.
- Indicar cuantos datos van a ser utilizados en cada *split* (*train*, *test* y *val*).

En experimentos

- Permitir ejecutar un modelo varias veces seguidas, a fin de generar varias versiones del modelo con las cuales comparar.

En resultados

- Incluir visualizaciones de curvas de error y métricas de precisión.
- Incluir gráficos para comparar distintos modelos.

- Permitir exportar el modelo.
- Comparar distintas ejecuciones (versiones) para un mismo modelo.

Antes de evaluar la interfaz en su versión horizontal, se realizó una prueba preliminar para detectar y corregir errores durante su fase de desarrollo. Esto permitió asegurar la funcionalidad de la aplicación antes de proceder con la prueba de usabilidad de la versión horizontal, tal como se presentará en la siguiente sección.

3.4 Pruebas Exploratorias

Durante las pruebas exploratorias se emplearon las heurísticas de Nielsen²⁴ como una guía fundamental para evaluar la usabilidad de la interfaz. Estas sirvieron para delimitar y categorizar los resultados obtenidos, identificando claramente las áreas de mejora y promoviendo una comprensión precisa de los aspectos que requieren atención y potenciales mejoras.

El objetivo de esta prueba fue identificar problemas en la interfaz y recopilar recomendaciones para mejorar la usabilidad usuaria en la versión horizontal y posteriores de DashAI.

3.4.1 Diseño

El método descrito se aplicó a cinco individuos, sin importar su experiencia previa en inteligencia artificial, sino que estuviesen familiarizados con la evaluación de interfaces. Los perfiles de los usuarios incluyeron dos estudiantes que han cursado asignaturas de interfaces de usuario y tienen conocimientos leves en *machine learning*, un antiguo ayudante de la asignatura de interfaces de usuario, un desarrollador de *software* especializado en interfaces, y un desarrollador de videojuegos con experiencia en la creación de interfaces para videojuegos.

La evaluación se realizó de manera remota utilizando AnyDesk, una aplicación que permite la transmisión e interacción sobre una pantalla compartida, facilitando así la conexión de los evaluados al computador del evaluador y evitando la necesidad de instalar las dependencias requeridas para ejecutar la aplicación de DashAI.

Una vez fue confirmada la viabilidad de la interacción entre las computadoras, se proporcionó a los usuarios una hoja de cálculo a través del chat compartido, el cual se puede consultar en el Anexo D.1. Este archivo contenía diez filas en donde cada una correspondía a una heurística y seis columnas, en donde se presentaron las secciones que debían ser rellenadas según lo solicitado. Estas fueron:

- Problema identificado: Debe indicar cual fue el error o insatisfacción encontrado en la aplicación.

²⁴ Publicadas por Jakob Nielsen en 1994 en el sitio web <https://www.nngroup.com/articles/ten-usability-heuristics/>, las heurísticas de Nielsen son diez reglas generales que pueden seguirse para crear productos digitales más accesibles y fáciles de usar.

- Componentes afectados: Que elemento o funcionalidad de la interfaz presento ese problema.
- Ejemplo o evidencia: Una imagen que muestre claramente el componente que el usuario identifico.
- Severidad: Una calificación baja, media o alta para indicar cuanto considera que ese problema le dificulta utilizar la aplicación.
- Recomendaciones o sugerencias: Indicar de qué manera se podría solucionar el problema.
- Comentarios generales: Cualquier observación o consideración adicional.

Los sujetos determinaron a qué heurísticas se refiere cada problema encontrado, pero la validación final fue realizada por el evaluador de la prueba. Para facilitar el proceso de identificación de heurísticas, se mostró un resumen de cada una en el archivo entregado.

Se les explicó a los usuarios las funciones de la aplicación y una breve descripción de cada módulo. También se les guio a través de un ciclo básico desde la carga de datos hasta la visualización de los resultados del modelo, todo lo anterior de manera verbal, sin mostrar elementos visuales, pero asegurándose de que comprendieran cada paso del proceso.

Al ser las pruebas exploratorias un método sin estructura definida, las heurísticas proporcionaron una guía efectiva para identificar las áreas de mejora y clasificarlas, de modo que el equipo de desarrollo visualice los puntos débiles de la interfaz, tales como el manejo de errores, la distribución de elementos, el tamaño de las imágenes, funciones de los botones, entre otros.

3.4.2 Ejecución

Cada uno de los problemas identificados por los sujetos fueron recopilados y se pueden encontrar en el Anexo D.2, junto con la cantidad de usuarios que lo notificaron, la severidad promedio asignada y la heurística otorgada.

A continuación, se detallan las distintas heurísticas que fueron afectadas por los problemas, de qué forma el estado actual de la interfaz afecta a esta heurística, así como recomendaciones para el equipo de desarrollo:

1. Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores:
 - Los mensajes de error son ambiguos o poco informativos, lo que dificulta que los usuarios comprendan el problema y cómo resolverlo.
 - Recomendación: Mejorar la claridad y relevancia de los mensajes de error, proporcionando sugerencias específicas para resolverlos.
2. Prevención de errores:

- La interfaz permite que los usuarios realicen acciones que pueden llevar a errores, como eliminar información importante sin una confirmación.
- Recomendación: Implementar medidas preventivas, como solicitar confirmación antes de acciones destructivas y ofrecer la opción de deshacer cambios.

3. Control y libertad del usuario:

- La aplicación limita las opciones del usuario en ciertos contextos, lo que puede generar frustración y sensación de falta de control.
- Recomendación: Brindar al usuario mayor libertad y control sobre sus acciones, permitiendo deshacer acciones, personalizar la interfaz y navegar fácilmente entre secciones.

4. Consistencia y estándares:

- La interfaz no sigue patrones de diseño ni convenciones de la plataforma u otros sistemas que los usuarios estén acostumbrados, lo que causa confusión y dificulta su familiarización.
- Recomendación: Adoptar patrones de diseño conocidos²⁵ y mantener la coherencia en la interfaz para que los usuarios puedan predecir el comportamiento de la aplicación.

5. Flexibilidad y eficiencia de uso:

- La aplicación no ofrece opciones de personalización ni atajos que faciliten la realización de tareas de manera eficiente. Además, carece de funciones adicionales que podrían agilizar la ejecución de las actividades de los usuarios.
- Recomendación: Implementar características que permitan a los usuarios personalizar la interfaz y agilizar su flujo de trabajo.

3.4.3 Recomendaciones

Los resultados se organizaron en los módulos correspondientes de DashAI. Se informó a los desarrolladores sobre la heurística asociada a cada mejora identificada, para que puedan consultar la literatura y encontrar recomendaciones de autores expertos para abordar los problemas asociados.

A continuación, se presentan las recomendaciones encontradas para los módulos de carga de datos, experimentos y resultados, considerando también la experiencia general del usuario al utilizar la aplicación.

²⁵ Tales como el diseño responsivo, el *scroll* infinito, las aplicaciones de una sola página, los componentes modulares, entre otros.

Carga de datos

- Visualizar cuantos elementos se ven afectados en cada *split*.

Experimentos

- Añadir un botón para visualizar los resultados de un experimento tras ejecutarlo.
- Indicar de antemano qué métricas serán visualizadas en los Resultados.

Resultados

- Permitir exportar un modelo.
- Utilizar un único ícono a la vez, evitar superponer íconos o juntar dos en uno.

General

- Cuando suceda un error, indicar claramente qué pasó y cómo solucionarlo
- Redirigir hacia la página principal tras pulsar el ícono de DashAI.
- Añadir un botón en todas las tablas de la interfaz que permita mostrar u ocultar cualquiera de las columnas presentadas en la vista.

Luego de detectar los errores y mejoras potenciales de la versión horizontal, se procedió a realizar una nueva prueba de usabilidad sobre esta versión corregida.

3.5 Prueba de usabilidad para la versión horizontal

La versión horizontal de la aplicación representa una mejora con respecto a la anterior. Se han incorporado algunas recomendaciones surgidas durante la primera y segunda iteración, en particular, aquellas derivadas de la comparación competitiva. Entre las características añadidas se incluye la posibilidad de subir varios archivos simultáneamente, seleccionar la tarea a desarrollar, crear múltiples experimentos y generar varios modelos por cada uno. Además, se ha implementado un módulo de resultados, que proporciona todos los parámetros utilizados y las métricas obtenidas por cada modelo. Es importante destacar que, en comparación con la primera versión, se ha eliminado la opción de insertar nuevos valores para que los usuarios prueben los modelos.

Para facilitar la comprensión de la interfaz utilizada, se presentará en la Figura 9, Figura 10, Figura 11 y Figura 12 los diferentes módulos de la aplicación.

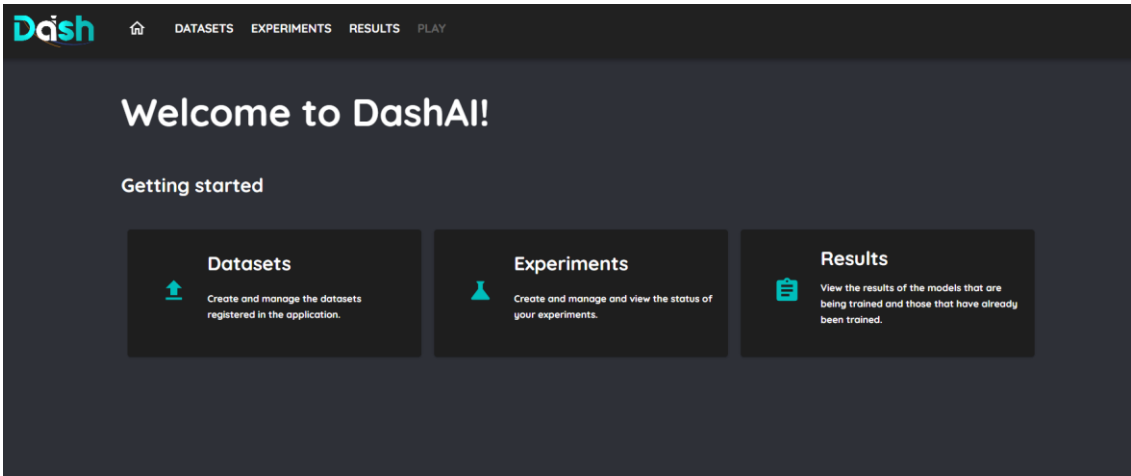


Figura 7: Página de inicio de la versión horizontal.

Fuente: Aplicación de DashAI.

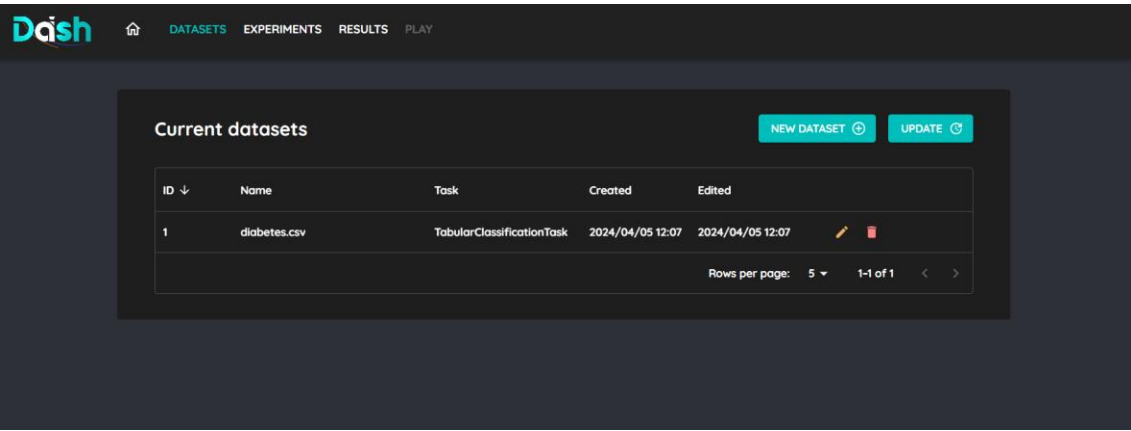


Figura 8: Módulo de carga de datos de la versión horizontal.

Fuente: Aplicación de DashAI.

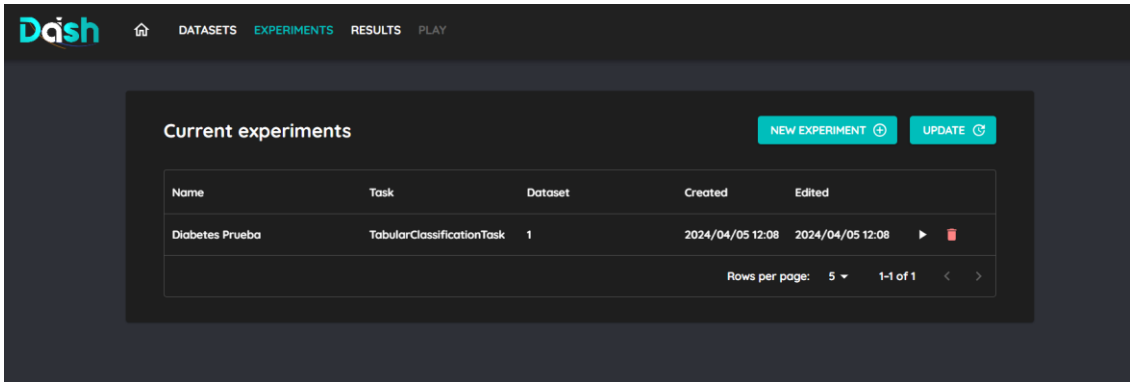


Figura 9: Módulo de experimentos de la versión horizontal.

Fuente: Aplicación de DashAI.

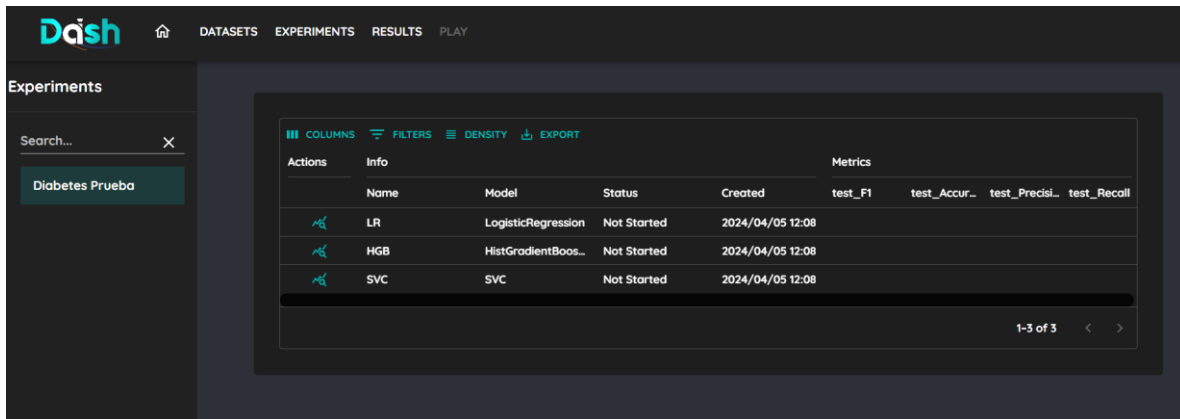


Figura 10: Módulo de resultados de la versión horizontal.

Fuente: Aplicación de DashAI.

El propósito de esta prueba de usabilidad fue detectar problemas en el flujo de la aplicación, recabar comentarios sobre la interfaz y su disposición de elementos para presentar los distintos módulos y funcionalidades, y obtener posibles mejoras tanto para los elementos actuales como para los que los usuarios desearían tener en versiones futuras.

3.5.1 Diseño

Este método fue aplicado a cinco personas con diferentes niveles de conocimiento en torno al *machine learning*:

- Un estudiante de pregrado con dos años de experiencia en inteligencia artificial.
- Un estudiante autodidacta de pregrado con más de un año de experiencia en proyectos relacionados con IA.
- Un profesional con más de dos años de experiencia como analista de datos en una compañía aseguradora de vehículos.
- Un arquitecto de software con más de dos años de experiencia en el análisis de modelos para su inclusión en aplicaciones.
- Un estudiante de postgrado en inteligencia artificial con más de tres años de experiencia en el campo.

Las pruebas se llevaron a cabo utilizando AnyDesk. Una vez verificada la conexión e interacción con la aplicación, se notificó al usuario la solicitud de grabar la sesión, indicándole que en cualquier momento puede solicitar la eliminación de fragmentos o la totalidad de la grabación.

El flujo de la reunión inició primero con un saludo, seguido de una introducción a DashAI y el propósito de la prueba. Posteriormente, se llevó a cabo una entrevista inicial para recopilar información sobre la experiencia en el área, seguida por la ejecución de distintas

tareas de usuario y, finalmente, una entrevista final sobre la experiencia en la interfaz. Además, se les solicitó a los participantes que completaran una serie de preguntas en formato de encuesta para recopilar información adicional que podría ser utilizada para futuras versiones.

Centrándose en el contenido de la prueba de usabilidad, se evaluaron los aspectos de la interfaz mediante cuatro tareas específicas. Estas tareas se enfocaron en distintas funcionalidades, como la carga de datos, la ejecución de un experimento, la visualización de los resultados de un experimento y la comparación de los resultados entre distintos modelos de un experimento. Las tareas específicas que los usuarios desarrollaron se detallan en la Tabla 3.

Tabla 4: Tareas que el usuario realizó en la prueba de usabilidad de la versión horizontal.

Fuente: Elaboración propia.

ID	Tarea	Descripción	Requerimientos
T1	Subir un archivo al sistema.	El usuario entiende cómo subir un archivo.	Archivo: diabetes.csv
T2	Crear un experimento.	El usuario entiende cómo generar un modelo para hacer predicciones.	Elección del modelo a utilizar.
T3	Ver los resultados detallados de un experimento.	El usuario visualiza la información detallada de un modelo utilizado en un experimento.	Crear un experimento.
T4	Comparar dos o más modelos.	El usuario logra comparar modelos de un experimento.	Crear un experimento con dos o más modelos.

Una vez finalizadas todas las pruebas, se procedió a realizar un resumen de cada una de las reuniones, en el cual se registraron minuciosamente todas las recomendaciones y comentarios identificados durante el proceso. Estas observaciones fueron posteriormente recopiladas y consolidadas en un informe detallado que se presentará como recomendaciones generales de esta prueba de usabilidad. El documento utilizado de guía por el entrevistador se puede encontrar en el Anexo E.1.

3.5.2 Ejecución

Se registró el tiempo que cada usuario tardó en completar cada tarea, como se detalla en la Tabla 4. El Usuario 1 se demoró considerablemente más en la tarea T4, debido a que se dedicó a inspeccionar el *dataset* y probar distintos filtros en las columnas ofrecidas.

Tabla 5: Tiempo en segundos que se demoró cada usuario en completar las tareas de la prueba de usabilidad de la versión horizontal.

Fuente: Elaboración propia.

Usuario/Tarea	T1	T2	T3	T4
Usuario 1	102	85	100	385
Usuario 2	179	127	84	210
Usuario 3	169	153	159	88
Usuario 4	70	66	140	122
Usuario 5	106	75	88	156

Durante la ejecución de cada tarea, se registraron los comentarios directos de los participantes junto con las observaciones del entrevistador. De estos registros, surgieron una serie de patrones importantes que se mencionan a continuación:

- Algunos usuarios necesitaron consultar el archivo a utilizar para saber qué tipo de separador (coma o punto coma) se iba a utilizar, mientras que los otros asumieron que era por coma.
- En general los usuarios mostraron confusión al no saber cuál fue la variable objetivo utilizada para realizar la predicción, lo que generó instancias de dudas durante el desarrollo, consultando para ello nuevamente el archivo cargado y exponiendo su incertidumbre al entrevistador.
- Los sujetos manifestaron que la descripción de los parámetros y otros valores presentes en la interfaz eran insuficientes, ya que no les permitieron reconocerlos ni entender su función o el efecto de sus modificaciones.

Los comentarios detallados recopilados se pueden encontrar en el Anexo E.2.

Por último, durante la entrevista final se recopiló información sobre diversas características de la aplicación mediante una serie de preguntas. A continuación, se resumen las respuestas de los usuarios:

- La mayoría encontró la interfaz fácil e intuitiva.

- Todos pudieron completar las tareas sin dificultad alguna.
- Algunos botones y detalles de los experimentos no se entendieron o visualizaron adecuadamente.
- Les gustaría que hubiera gráficos y que se permitieran explorar los datos antes de ser cargados.
- Que en alguna sección se visualice la variable objetivo o que se pueda seleccionar.
- Mejorar la descripción de los parámetros. Qué es lo que hacen y qué modifican.

Las respuestas anteriores fueron extraídas de las distintas preguntas realizadas, las cuales se detallan en el Anexo E.3. En este anexo, se pueden encontrar las nueve preguntas formuladas.

3.5.3 Recomendaciones

Luego de analizar la información obtenida, se han generado las siguientes sugerencias para los módulos de DashAI.

En la carga de datos

- Establecer un color particular para cada tipo de tarea diferente.
- Modificar las palabras para que tengan una lectura más humana (ej: cambiar de *TabularClassificationTask* a *Tabular Classification*).
- Visualizar algunos datos del archivo cargado.
- Indicar el progreso de carga de un archivo de datos.

En experimentos

- Añadir el modelo de redes neuronales.
- Agrandar el botón de iniciar experimento.
- Añadir una barra de progreso de un experimento.
- Permitir cancelar un experimento en ejecución.
- Agregar botón para cerrar la ventana asociada a la ejecución de un experimento.
- Permitir seleccionar la variable objetivo.
- Al finalizar un experimento, indicar claramente qué se predice.
- Modificar el ícono de actualizar los experimentos.
- Permitir subir un modelo pre-entrenado en formato Keras.

- Añadir un botón para pasar rápidamente de experimentos a resultados tras ejecutar un experimento.

En resultados

- Indicar a qué corresponde el número “*status*” de un experimento.
- Si un parámetro es *null* o no tiene valores asociados, ocultarlo.
- Visualizar qué elementos en específico el modelo falló en predecir.
- Permitir validar el modelo con nuevos datos.
- Mostrar una explicación de los parámetros en la vista general y detallada de los experimentos.
- Indicar sobre qué variable se realizó la predicción.
- El menú izquierdo no se ve claramente, se confunde con el fondo.
- Mostrar un gráfico de dispersión y una matriz de correlación.
- Añadir la métrica de sensibilidad por defecto.
- Agrupar las métricas de *test*, *train* y *val* en el detalle de los experimentos.
- Los experimentos con error deben indicar claramente qué pasó y cómo solucionarlo.

En la experiencia general al utilizar la aplicación

- Diseñar un nuevo módulo que muestre un ejemplo de ejecución completo de la aplicación.
- Tener cargados de manera predeterminadas algunos archivos de prueba con datos.
- Tener experimentos de prueba basados en archivos de prueba.
- Entregar una explicación básica de qué es lo que hace la plataforma

Finalizada la recopilación de las distintas recomendaciones procedentes de los métodos implementados en la primera y segunda iteración, se hizo necesario exponer los hallazgos descubiertos antes de comenzar la elaboración del prototipo. De este modo, se pudo identificar claramente en qué módulos debía enfocarse.

3.6 Resumen de hallazgos

Todos los hallazgos encontrados fueron divididos en sus módulos respectivos, además de contar con la experiencia general del usuario al utilizar la aplicación.

Por la parte de la carga de datos, se encontró que una solicitud recurrente por parte de los usuarios fue la necesidad de obtener un detalle más específico de la información cargada. Esto incluye la posibilidad de ver resúmenes estadísticos, manipular los datos mediante técnicas de normalización y centralización, y seleccionar qué atributos serán evaluados en el *dataset* cargado.

En los experimentos se observó que los usuarios sentían la falta de un botón para ver los resultados inmediatamente después de ejecutar un experimento, ya que tenían que moverse al módulo respectivo, lo que les quitaba tiempo. Además, no conocían la variable objetivo del experimento, lo cual les generaba incertidumbre sobre qué se estaba prediciendo. Los usuarios expresaron que les gustaría poder modificar la variable objetivo o, al menos, visualizar cuál es. También se destacó la necesidad de tener la opción de detener un experimento en ejecución, ya que actualmente no era posible pausarlos y había que esperar a que se completaran.

En los resultados, los usuarios querían saber la cantidad de elementos usados y cuáles fueron clasificados correcta o incorrectamente durante la validación. Algo muy solicitado fue la inclusión de gráficos para comparar distintos modelos entre sí y para evaluar el rendimiento de un único modelo. Una solicitud menos frecuente, pero importante, fue la de permitir exportar el modelo generado para ser replicado en otros ambientes.

Por último, en la experiencia general del usuario en la aplicación, se identificó la necesidad de mejorar la explicación de parámetros y métricas en todos los módulos, para que los usuarios puedan interpretar mejor lo que están haciendo. También se sugirió incluir tutoriales y explicaciones que clarifiquen el propósito de la plataforma y cómo utilizar cada módulo de la aplicación. Además, se recomendó agrandar los botones de los experimentos, en especial el de ejecutar, que consideran fundamental y cuyo tamaño actual no refleja su importancia.

Lo anterior fue un resumen de los hallazgos obtenidos. Para consultar la lista completa se puede ir al Anexo F.1 para ver los asociados a la carga de datos y experimentos, mientras que los referentes a los resultados y la experiencia general del usuario en el Anexo F.2.

Después de considerar las recomendaciones y discutir con el equipo cuáles se consideraron necesarias de implementar para el prototipo, se tomó la decisión de concentrarse en el módulo de resultados, específicamente en las recomendaciones relacionadas con los gráficos para comparar modelos y la mejora de la distribución de los elementos en la interfaz. Respecto a la primera, se decidió implementar los gráficos, dado que en la mayoría de los métodos los usuarios expresaron una necesidad de visualizar comparaciones gráficas de los modelos. Esta característica no estaba presente en DashAI, y su incorporación resultaría innovadora para la plataforma. En cuanto a la segunda, el equipo decidió reestructurar la información del módulo de resultados. Este módulo tenía una sección particular en la aplicación, pero se integraría dentro del módulo de experimentos. Los resultados serían accesibles mediante un botón para cada uno de los

experimentos, de modo que la información quedara más resumida y los usuarios no tuvieran que desplazarse a otro módulo.

Luego de tener todo claro respecto a las modificaciones a realizar en el prototipo, se procedió a diseñarlo en Figma, tal como se muestra en la siguiente subsección.

3.7 Prototipo

En esta nueva versión, para acceder a los resultados de un experimento se debe pulsar un botón triangular a la derecha de un experimento ya ejecutado. Esto abrirá una ventana que muestra información sobre los modelos, incluyendo nombre, algoritmo usado, estado de ejecución y métricas. En la parte superior de la ventana aparecerán dos botones: "Columns", que permite ver la vista actual, y "Graphs", que lleva a una nueva vista con gráficos comparativos. La Figura 11 muestra una comparación entre la versión horizontal (arriba) y la versión prototipo (abajo), destacando el cambio de apariencia.

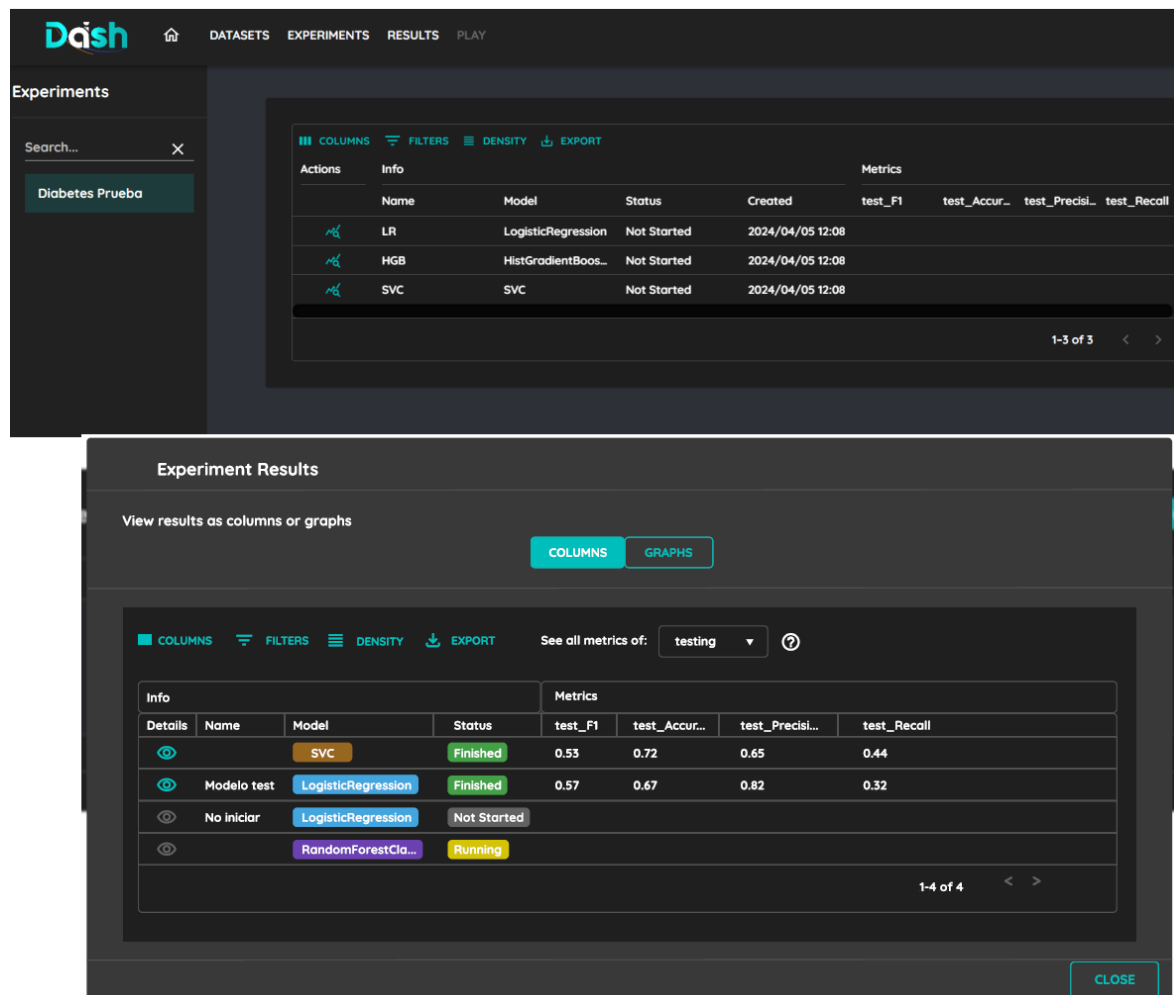


Figura 11: Comparación entre el módulo de resultados de la versión horizontal con la versión prototipo.

Fuente: Elaboración propia.

Una vez se accede a la vista de gráficos, aparecerá la opción de elegir qué gráfico mostrar. El gráfico interactivo seleccionado se ubicará en el centro de la pantalla, y a la izquierda estará la opción de configurar qué métricas utilizar para la comparación.

Se destacan dos formas de visualización de métricas:

- **Métricas específicas:** Se muestran todas las métricas posibles para comparar, incluyendo F1, *accuracy*, *precision* y *recall*. Cada una de estas métricas se repite en los contextos de *training*, *testing* y *validation*, sumando un total de doce métricas diferentes. El usuario puede elegir específicamente qué métricas comparar. Para que aparezca un gráfico, se debe tener al menos una métrica específica seleccionada. Un ejemplo de esto se muestra en la Figura 12.
- **Métricas generales:** Agrupan las métricas según sus nombres. Por ejemplo, la métrica general "*testing*" mostrará todas las métricas de *testing* (*accuracy*, *precision*, *recall* y F1), mientras que la métrica "F1" mostrará las métricas *training* F1, *testing* F1 y *validation* F1. Esto permite al usuario comparar directamente las métricas sin tener que seleccionarlás individualmente en la vista de métricas específicas. Un ejemplo de esto se muestra en la Figura 13.

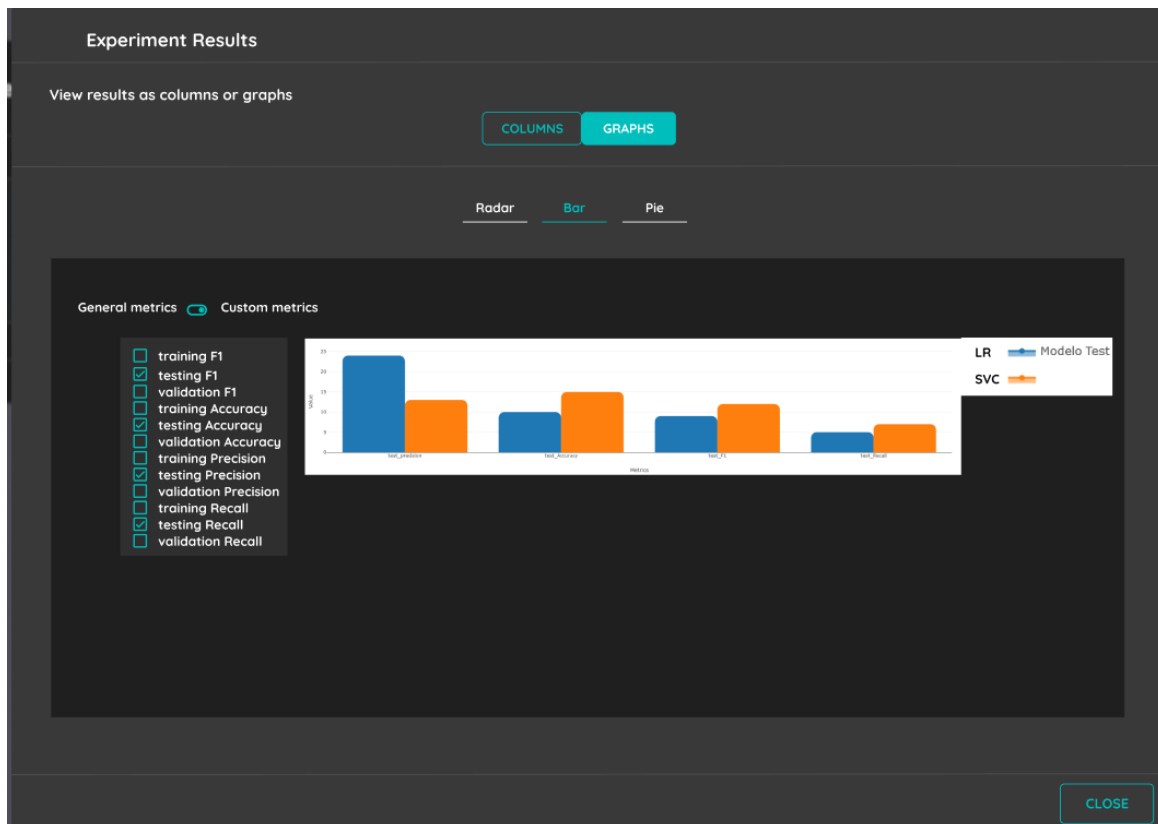


Figura 12: Prototipo de la elaboración de los gráficos para comparar los modelos, para cualquiera de las métricas.

Fuente: Elaboración propia.

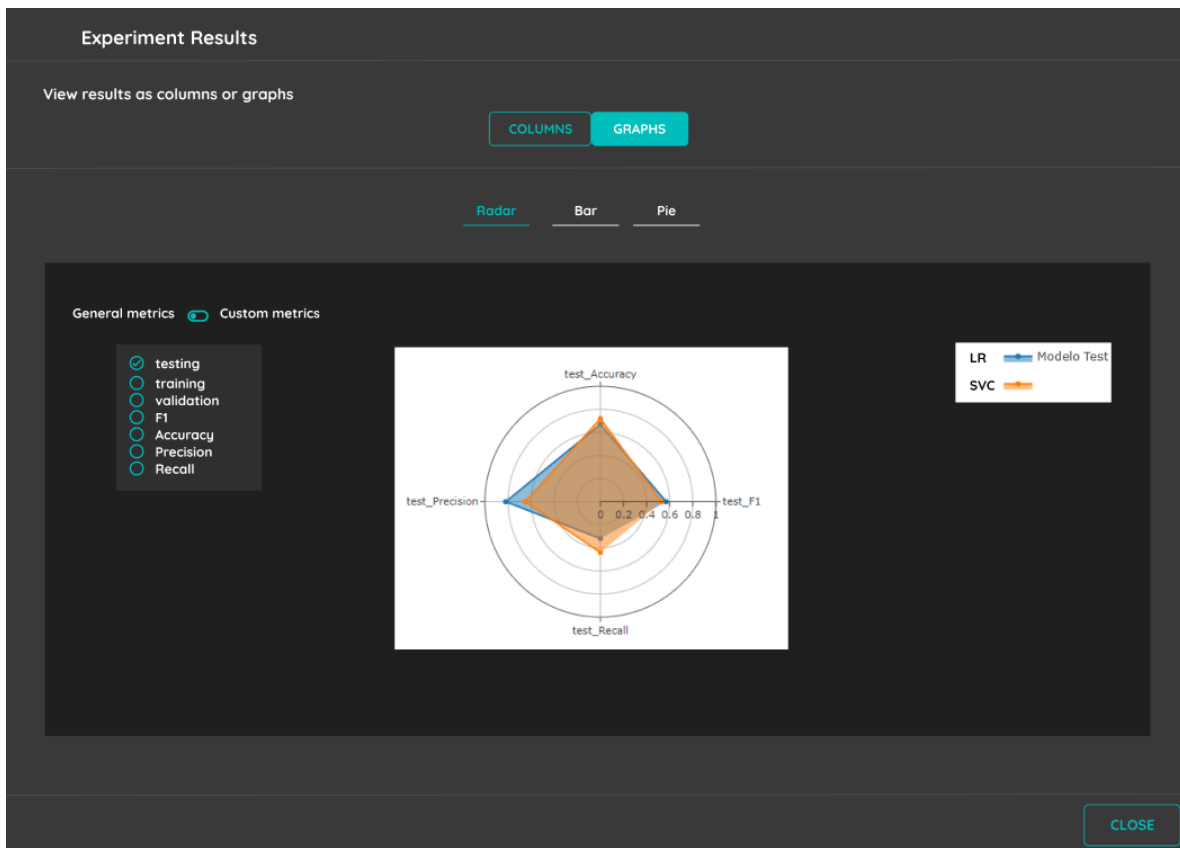


Figura 13: Prototipo de la elaboración de los gráficos para comparar los modelos, para las métricas generales.

Fuente: Elaboración propia.

La decisión de implementar las métricas generales no se derivó de las recomendaciones obtenidas mediante los métodos utilizados, sino que se elaboró con el único fin de disminuir la cantidad de clics que el usuario debe realizar para poder comparar.

Tras generar el prototipo en Figma y revisarlo con el equipo en varias reuniones diarias y semanales, se validó y fue aprobado para proceder a su implementación en código.

3.7.1 Implementación

Debido a las mejoras en curso en distintas ramas, se creó una denominada "front/results-plot" para su desarrollo. Una vez que todo el contenido estuvo listo, se envió una *Pull Request*, a la espera de que se vincule a la versión principal, tal como se puede observar en la Figura 14.

Front/results plot #196

 Open danielroco wants to merge 16 commits into `develop` from `front/results-plot` 

 Conversation 0

 Commits 16

 Checks 8

 Files changed 49



danielroco commented 2 weeks ago • edited ▾

Collaborator ...

Summary

Elimination of the Results module for integration into the Experiments section has been implemented. The presentation of the Runs table and its details has been refined, alongside the incorporation of data visualization through Plotly. Currently supported chart types include Radar, Bar, and Pie Charts.

Type of change

- Front end new feature.
- Bug fix.
- Refactoring.

Changes

- Removal of the Results module and all corresponding calls to this module.
- The Results module has been relocated to a Dialog within the Experiments module.
- Optimization of the code for calling the Results module.
- Refactoring of the entire Results section.
- Implementation of charts using Plotly, currently supporting Radar, Bar, and Pie Charts.
- Generalization in data retrieval for the charts.
- Allow the user to choose which metrics to display on the charts or utilize the general ones.
- Frontend adjustments based on usability testing.
- Addition of alerts in various parts of the interface.

Outstanding

Outstanding refactors:

- ResultsTable.jsx

Figura 14: *Pull Request* del prototipo implementado.

Fuente: Repositorio GitHub del proyecto.

Luego de numerosas validaciones por parte de los desarrolladores *senior*, se implementó en la versión estable de la aplicación. En esta versión también se incluyeron otros cambios realizados por otros desarrolladores, pero estos no afectaron las mismas secciones modificadas en el prototipo. Por esta razón, se decidió realizar la prueba de usabilidad en la *branch* de la versión estable “*develop*”, prueba que se detallará en el siguiente capítulo.

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN

En este capítulo se aborda si la implementación de las mejoras basadas en las experiencias usuarias ha contribuido a mejorar DashAI de alguna manera. Esto se logra mediante el análisis de los resultados de la última prueba de usabilidad, la evaluación de un cuestionario de usabilidad sobre la experiencia, y una comparación entre el estado previo al desarrollo de este trabajo y el estado posterior, que corresponde al término de la cuarta iteración.

4.1 Prueba de usabilidad del prototipo final

En esta versión, se introduce el rediseño propuesto junto con otras funcionalidades desarrolladas. Entre los cambios importantes se destacan la eliminación de la selección de *task* al cargar los datos, la visualización de un resumen de los datos cargados, la capacidad de establecer una variable objetivo, la integración del módulo resultados en experimentos mediante un botón, y la implementación de los gráficos definidos en el prototipo.

El propósito de esta prueba de usabilidad es obtener una retroalimentación sobre la usabilidad de la aplicación, centrándose en el módulo de resultados. El objetivo es verificar si la inclusión de los gráficos añade valor para los usuarios, determinando si es apropiado mantenerlo o no, y sugiriendo posibles modificaciones.

4.1.1 Diseño

Este método se aplicó a cinco personas que hayan participado previamente en alguna de las pruebas de usabilidad, cuyo rango de experiencia varía entre cero y tres años. Los participantes fueron seleccionados en función de su participación proactiva y disponibilidad.

El utilizar usuarios previos tiene varias ventajas:

- **Familiaridad con la aplicación:** Los participantes que ya han interactuado con la aplicación en versiones anteriores poseen una comprensión básica de su funcionalidad y flujo de trabajo. Esto minimiza el tiempo y esfuerzo necesarios para familiarizarse con la aplicación durante la prueba de rediseño, permitiendo que se enfoquen en evaluar los cambios específicos realizados.
- **Comparación:** Al haber experimentado versiones anteriores, estos usuarios pueden comparar directamente su experiencia anterior con la nueva versión rediseñada. Esta comparación facilita la identificación de mejoras, áreas problemáticas y cambios generales en la usabilidad.
- **Eficiencia en reclutamiento:** Reutilizar participantes de pruebas anteriores reduce el tiempo y esfuerzo necesarios para reclutar nuevos usuarios, especialmente si el grupo objetivo es limitado o especializado.

Es importante tener en cuenta algunas consideraciones al utilizar usuarios previos:

- Selección cuidadosa de participantes: Se debe considerar su nivel de participación en pruebas anteriores, su diversidad en términos de características demográficas y patrones de uso, y su entusiasmo por proporcionar comentarios útiles.
- Enfatizar los aspectos rediseñados: Durante la prueba, se debe enfatizar claramente a los participantes qué aspectos de la aplicación han sido rediseñados y qué se espera que evalúen. Esto ayudará a dirigir sus comentarios hacia los cambios específicos introducidos en la nueva versión.
- Sesgos: Es importante ser consciente del potencial de sesgos en la retroalimentación de usuarios con experiencia previa. Para mitigar esto, se deben utilizar métodos de recolección de datos que fomenten la honestidad y la objetividad, como encuestas anónimas o entrevistas en profundidad.

Las pruebas se llevaron a cabo utilizando AnyDesk, lo que facilitó la conexión de los evaluados al equipo del evaluador y evitó la necesidad de instalar las dependencias necesarias. Una vez que se haya verificado la conexión e interacción con la aplicación, se notificó al usuario la solicitud de grabar la sesión, indicándole que en cualquier momento puede solicitar la eliminación de fragmentos o la totalidad de la grabación.

El flujo de la reunión constó primero de un saludo, seguido de una introducción a DashAI y el propósito de la prueba. Posteriormente, se llevó a cabo una entrevista inicial para recopilar información sobre la experiencia en la prueba previa, seguida por la ejecución de distintas tareas de usuario y, finalmente, se les solicitó a los participantes que completen un cuestionario de usabilidad, el cual fue anónimo.

Centrándose en el contenido de la reunión, se evaluaron los aspectos de la interfaz para la versión prototipo de la aplicación mediante cuatro tareas específicas. Estas tareas se enfocaron en distintas funcionalidades de la interfaz, como la carga de datos, la ejecución de un experimento, la visualización de los resultados de un experimento y la comparación de los resultados entre distintos modelos de un experimento, centrándose en las dos últimas. Las tareas específicas que los usuarios desarrollaron se detallan en la Tabla 6. Cabe mencionar que, en la tercera tarea, siempre apareció la visualización de tabla como vista predeterminada.

Una vez finalizadas todas las pruebas, se procedió a realizar un análisis sobre todas las respuestas del cuestionario, lo cual fue fundamental para extraer conclusiones significativas sobre la experiencia del usuario con las nuevas funcionalidades implementadas en la aplicación.

Tabla 6: Tareas que el usuario realizó en la prueba de usabilidad de la versión prototipo.

Fuente: Elaboración propia.

ID	Tarea	Descripción	Requerimientos
T1	Cargar un <i>dataset</i> y observar el resumen de datos.	El usuario entiende cómo se sube un archivo a la aplicación y observa el resumen de los datos.	Archivo con extensión .csv
T2	Crear y ejecutar un experimento.	El usuario entiende cómo generar varios modelos para hacer predicciones.	Elección de los modelos a utilizar.
T3	Ver los resultados.	El usuario visualiza la información en dos formatos distintos.	Crear un experimento.
T4	Comparar varios modelos.	El usuario logra comparar modelos de un experimento.	Crear un experimento con varios modelos.

Para el cuestionario de usabilidad final se empleó SUS con enunciados adaptados específicamente al contexto de modelado y análisis de resultados. La relación entre estos enunciados originales y los adaptados se detallan en la Tabla 7.

Tabla 7: Adaptación de las preguntas SUS en DashAI.

Fuente: Elaboración propia.

Pregunta SUS	Pregunta adaptada de DashAI
Creo que me gustaría utilizar este sistema con frecuencia	Me gustaría utilizar esta interfaz con frecuencia debido a su atractivo visual y experiencia agradable
Encontré el sistema innecesariamente complejo	La navegación desde la carga de datos hasta los resultados me pareció innecesariamente compleja
Pensé que el sistema era fácil de usar	Pensé que la interfaz hacía fácil acceder a toda la información necesaria una vez que los datos estaban cargados
Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema	La interfaz está diseñada exclusivamente para usuarios expertos.

Encontré que las diversas funciones de este sistema estaban bien integradas	Encontré que las funciones de visualización de gráficos y comparación de modelos estaban bien integradas.
Pensé que había demasiada inconsistencia en este sistema	Hubo inconsistencia en la explicación de ciertos elementos, como parámetros, modelos y métricas
Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente	Me imagino que la mayoría de la gente aprendería a utilizar esta interfaz rápidamente.
Encontré el sistema muy complicado de usar	La comparación de los modelos resultó complicada
Me sentí muy seguro usando el sistema	Me sentí seguro usando esta interfaz
Necesitaba aprender muchas cosas antes de empezar con este sistema	Necesitaba aprender muchas cosas antes de empezar a personalizar elementos como gráficos, parámetros y modelos

El documento utilizado de guía por el entrevistador se puede encontrar en el Anexo G.1.

4.1.2 Ejecución

Se registró el tiempo que cada usuario tardó en completar cada tarea, como se detalla en la Tabla 8. Para la tarea T2, referente a crear y ejecutar un experimento, todos los usuarios se demoraron más que en las pruebas de usabilidad anteriores, debido a la implementación de la opción de fijar una variable objetivo, algo que les gustó bastante y los llevó a investigar más a fondo esta nueva funcionalidad.

Tabla 8: Tiempo en segundos que se demoró cada usuario en completar las tareas de la prueba de usabilidad de la versión final del prototipo.

Fuente: Elaboración propia

Usuario/Tarea	T1	T2	T3	T4
Usuario 1	75	142	103	91
Usuario 2	103	136	65	188
Usuario 3	128	235	60	87
Usuario 4	135	226	154	174
Usuario 5	112	211	88	144

Es importante destacar que todos los usuarios pudieron completar todas las tareas encomendadas.

Durante la ejecución de cada tarea, se registraron los comentarios directos de los participantes junto con las observaciones del entrevistador. De estos registros, surgieron una serie de patrones importantes que se mencionarán a continuación. Cabe recordar al lector que en esta prueba también se evaluaron cambios realizados en la versión por otros desarrolladores.

4.1.3 Hallazgos encontrados

La mayoría de las opiniones realizadas por los usuarios correspondieron a problemas identificados o cambios que se deben realizar. Esto pudo deberse a que ya habían utilizado la aplicación anteriormente y decidieran no dar comentarios sobre algo que había funcionado en versiones pasadas.

En la carga de datos los usuarios manifestaron que sigue el problema de que no saben identificar si archivo cargado debe ser separado por coma o no, con lo cual es importante tener en cuenta que se debería implementar una detección automática del separador o que se mencione en alguna parte al cargar el archivo. En la versión evaluada se había implementado una tabla para visualizar los tipos de datos cargados (numérico, texto, booleano, entre otros), lo cual agrado a los usuarios, pero comentaron que preferirían otro tipo de visualización, la cual se enfocara en dar estadísticas sobre el *dataset*, tales como el promedio, mediana, desviación estándar, entre otros, considerando siempre y cuando calce con los tipos de datos.

En el módulo de experimentos surgió una confusión en las secciones de *split*. Al cargar un archivo, se le pregunta al usuario la división a realizar, pero en la vista en donde se selecciona la variable objetivo, también se le pregunta por una división de *split*, lo que provocó confusión sobre si era para ese experimento o para el *dataset*. Los usuarios también encontraron un error luego de modificar el valor del entrenamiento y validación en los *splits* por cualquier número, donde automáticamente la casilla quedaba remarcada en rojo y la marca no desaparecía, cuando no debería haberse mostrado. Otra cosa señalada es que a la mayoría no les gustó que se mostrara la variable objetivo como el número de columna que la identifica en el archivo, prefiriendo que se indique el nombre de la variable en vez de un número.

Los hallazgos encontrados en el módulo de resultados, lugar donde se realizó la modificación basada en el prototipo, será evaluada en la siguiente subsección.

4.1.4 Hallazgos sobre lo modificado en el prototipo

Los usuarios no tuvieron problemas al encontrar el botón para poder acceder a los resultados mediante el módulo de experimentos, pero si comentaron que les gustaría que tuviera un tamaño mayor, dado la importancia que este presenta para el flujo de la aplicación.

Se valoró positivamente la implementación de los gráficos, considerándola útil y beneficiosa al proporcionar una visualización alternativa, permitiéndoles comparar los distintos modelos generados. Un punto fuerte que se encontró fue que eran interactivos, lo cual le gusto a los usuarios al poder hacer zoom, recortar secciones de los gráficos y poder decidir cuales modelos mostrar.

Un usuario comentó: “para alguien que desarrolla el modelo encontraría útil la tabla de resultados, mientras que para alguien que no esté familiarizado con ello, preferiría los gráficos”. Esto da una retroalimentación muy valiosa sobre las distintas utilidades que podrían tener los gráficos más allá de poder comparar los modelos.

Los gráficos presentados incluyeron radar, barra y torta. Este último causó confusión a la gran mayoría de los usuarios, quienes no entendieron el significado de cada sección del gráfico ni cómo podían comparar los distintos modelos para determinar cuál tenía el mejor rendimiento, lo que generó cuestionamientos sobre su utilidad. Mientras que varios usuarios mencionaron que el gráfico de barras sería el que más usarían debido a su familiaridad, otros comentaron que podrían utilizar el gráfico de radar, ya que su representación les pareció atractiva.

Durante la prueba, un participante no accedió a las métricas personalizadas mediante su botón, lo cual debe ser evaluado para asegurarse de que los futuros usuarios se den cuenta de esta funcionalidad. Se sugiere agrandar dicho botón o remarcarlo con un color distintivo. Además, algunos modelos en el gráfico de radar no pudieron ser comparados adecuadamente debido a que no se les había asignado un nombre. Esto será un tema de discusión con el equipo para determinar si el nombre de un modelo debe ser obligatorio o si debería seguir siendo opcional.

Por último, los usuarios sugirieron que en la sección de gráficos se incluya una explicación de todas las métricas mostradas, para que puedan interpretar mejor los resultados sin tener que consultar la documentación ante cualquier duda.

Todos los hallazgos encontrados en la prueba pueden ser consultados en el Anexo G.2.

4.1.5 Resultados del cuestionario

Los resultados del SUS fueron recopilados a través de formularios creados para este fin. Posteriormente, se sumaron las respuestas totales para cada pregunta según la respuesta dada, las cuales iban desde "Totalmente en desacuerdo" hasta "Totalmente de acuerdo", tal como se puede ver en la Figura 15, en donde los números indican cuantas personas marcaron esa respectiva opción en la pregunta dada.

Pregunta SUS	Totalmente en desacuerdo	En desacuerdo	Ni de acuerdo ni en desacuerdo	De acuerdo	Totalmente de acuerdo
Me gustaría utilizar esta interfaz con frecuencia debido a su atractivo visual y experiencia agradable.	0	0	0	4	1
La navegación desde la carga de datos hasta los resultados me pareció innecesariamente compleja.	2	3	0	0	0
Pensé que la interfaz hacía fácil acceder a toda la información necesaria una vez que los datos estaban cargados.	0	0	1	2	2
La interfaz está diseñada exclusivamente para usuarios expertos.	2	2	1	0	0
Encontré que las funciones de visualización de gráficos y comparación de modelos estaban bien integradas.	0	0	3	1	1
Hubo inconsistencia en la explicación de ciertos elementos, como parámetros, modelos y métricas.	1	1	2	1	0
Me imagino que la mayoría de la gente aprendería a utilizar esta interfaz rápidamente.	0	0	0	1	4
La comparación de los modelos resultó complicada.	3	2	0	0	0
Me sentí seguro usando esta interfaz.	0	0	1	4	0
Necesitaba aprender muchas cosas antes de empezar a personalizar elementos como gráficos, parámetros y modelos.	0	1	0	3	1

Figura 15: Resultados del SUS.

Fuente: Elaboración propia.

El puntaje obtenido del SUS fue de 73.5, lo cual está levemente por encima del promedio de 68. Dentro de la evaluación, esto considera que la aplicación tiene una usabilidad "aceptable". Sin embargo, debido al número limitado de participantes para este cuestionario, esta calificación no debería considerarse definitiva, por lo que se preferirá la evaluación cualitativa sobre la cuantitativa.

4.2 Comparación entre la situación previa y la actual

Es importante considerar no solo los resultados de la prueba de usabilidad, sino también toda la influencia ejercida a lo largo del proyecto. Mediante los distintos métodos aplicados, se obtuvo información valiosa sobre las necesidades de los usuarios. Esto es especialmente relevante considerando la situación inicial de DashAI, en la cual la aplicación no había sido probada exhaustivamente con usuarios y se desconocía si les gustaría o no. A medida que se avanzaba en el proyecto, se proporcionaron recomendaciones al equipo de DashAI, algunas de las cuales fueron implementadas, entre las que se encuentran:

- Permitir seleccionar *splits* de un experimento.
- Añadir una indicación visual sobre cuándo se cargó correctamente un *dataset*.
- Permitir modificar la semilla para generar la división en la carga de datos.
- Mostrar los primeros datos del *datasets* y columnas.
- Añadir una descripción de lo que hacen los modelos.
- Permitir elegir una variable objetivo.
- Establecer una configuración avanzada de los parámetros en los modelos.
- Permitir filtrar las columnas del *dataset* antes de un experimento.
- Añadir una descripción a cada parámetro.
- Establecer como máximo dos decimales para los valores numéricos.
- Validar los errores antes de continuar sobre otro componente.
- Enumerar las secciones de la interfaz y mostrar en dónde está el usuario.
- Añadir gráficos para comparar modelos.

Si bien se implementaron, aún hay aspectos por perfeccionar, como la descripción de los parámetros y la validación de los errores que surgen, problemas que reivindicaron su existencia en la prueba de usabilidad sobre el prototipo. Estas acciones reflejan un compromiso continuo con la mejora y optimización que debe tener el equipo de DashAI en beneficio de los usuarios finales.

CAPÍTULO 5: CONCLUSIONES

Antes de cerrar este informe, es importante destacar algunas conclusiones clave derivadas del desarrollo y las pruebas realizadas en el proyecto. Además, se identifican áreas que requieren atención adicional en futuras iteraciones para seguir mejorando la experiencia del usuario y el funcionamiento del sistema.

Durante el transcurso de este proyecto se realizaron pruebas que proporcionaron una visión clara de la experiencia del usuario con la aplicación. Se identificaron áreas de mejora en torno a los módulos, explicaciones, funcionalidades y visualización de los resultados. Es importante reconocer que la evaluación de la experiencia del usuario es un proceso continuo y subjetivo, y que los resultados obtenidos pueden variar según el contexto y las preferencias individuales de los usuarios. Por lo tanto, aunque los resultados proporcionan una indicación sólida del rendimiento de la aplicación en un momento dado, es importante seguir recopilando *feedback* y realizando evaluaciones periódicas para garantizar su relevancia y validez a lo largo del tiempo.

Es crucial resaltar la importancia de los comentarios que surgieron de los usuarios una vez que completaron las tareas asignadas o después de la entrevista final. Aunque los participantes respondieran de acuerdo con lo previsto durante el transcurso de la evaluación, muchos de ellos ofrecieron perspectivas valiosas y aportaron información relevante al concluir la reunión. Estas observaciones posteriores pueden revelar aspectos no detectados durante la prueba o proporcionar *insights* adicionales sobre la experiencia. Por lo tanto, mantenerse atento y ser proactivo en la recopilación de información, incluso después de que la prueba haya finalizado formalmente, puede ser crucial para identificar áreas de mejora y optimizar el diseño o funcionamiento del producto o servicio en desarrollo.

Existe una clara dificultad para encontrar participantes adecuados. Dado que el proyecto involucra inteligencia artificial, específicamente aprendizaje automático, localizar individuos que cumplan con los criterios del público objetivo puede resultar complicado. Una ventaja de llevar a cabo el proyecto mientras se está en la universidad es la oportunidad de conectar con personas que están cursando programas afines. Esta situación facilita la creación de una red de contactos, lo que a su vez permite acceder a nuevos usuarios potenciales para realizar pruebas y obtener retroalimentación sobre el producto en desarrollo.

La colaboración en equipo con desarrolladores que están continuamente implementando nuevas funcionalidades plantea una gran complejidad. A lo largo de las diversas versiones del proyecto, surgieron cambios que afectaron sustancialmente un modelo o función, lo que requirió la reescritura de las pruebas para adaptarlas a las modificaciones. Además, se encontró con la situación opuesta, donde había implementaciones pendientes que podrían haber generado cambios significativos. Idealmente, estas modificaciones habrían sido probadas, pero lamentablemente quedaron descartadas debido a la falta de tiempo.

para su implementación. Esta dinámica resalta la importancia de una comunicación eficaz y una planificación cuidadosa entre el equipo de desarrollo y los responsables de las pruebas, para asegurar que todas las funcionalidades se prueben adecuadamente antes de su lanzamiento.

La creación y mantenimiento de una documentación detallada a lo largo del proceso de rediseño ha sido esencial para garantizar la coherencia y la continuidad del proyecto. Esta práctica ha permitido a todos los miembros del equipo acceder a la información necesaria y entender el estado del proyecto en todo momento, facilitando una mejor toma de decisiones y una implementación más eficiente de los cambios.

Desde antes del inicio del proyecto de rediseño, algunos elementos y funcionalidades ya implementados fueron muy bien recibidos por los usuarios durante las diversas pruebas. Entre estos elementos se destaca la división de la interfaz en distintos módulos, que abarca desde la carga de *datasets* hasta la visualización de resultados. Los usuarios encontraron esta estructura muy intuitiva, y no hubo prácticamente ningún comentario negativo sobre la claridad del flujo. Además, la paleta de colores oscuros fue bien recibida, aunque algunos usuarios sugirieron ajustes leves en el color de ciertos íconos, pero manteniendo tonalidades similares. Estos elementos deberían mantenerse en futuras versiones, ya que las pruebas demostraron su aceptación positiva.

Por otra parte, hubo ciertos elementos y funcionalidades que generaron problemas en cada nueva versión. Entre estos se encuentran los errores que no incluían una clara solución, causando confusión en los usuarios. Los errores más graves reportados fueron el fallo en carga de *datasets* y la ejecución de experimentos, en los que les aparecía un mensaje de error, pero que no supieron como continuar para resolverlo, reiniciando para ello la aplicación. Además, se identificó la falta de una explicación clara sobre los parámetros a configurar y las métricas utilizadas. Muchos usuarios no entendieron su finalidad ni el impacto que tienen, ya que las descripciones se limitaban a indicar qué era, sin detallar qué hacían, de qué manera afectaba al modelo o sus rangos. Estos problemas deben corregirse antes de lanzar la aplicación, ya que podrían afectar negativamente su aceptación por parte de los usuarios.

Aunque DashAI presenta aspectos tanto positivos como negativos, siempre hay áreas donde se puede mejorar, incluso si el sistema funciona bien actualmente. Una mejora potencial sería la creación de un nuevo módulo de ejemplos que permita a los usuarios experimentar con la aplicación y comprender mejor su flujo de trabajo. Además, la introducción de tutoriales dentro de cada módulo podría resultar invaluable para usuarios que se encuentren atascados, especialmente aquellos que están aprendiendo *machine learning*, como estudiantes universitarios. Otra área de mejora podría ser la inclusión de herramientas para el análisis exploratorio de datos al cargar los *datasets*. Aunque los usuarios lograron cargar los archivos requeridos, muchos desearían haber podido explorar sus *datasets* con mayor profundidad.

A lo largo del proyecto se han generado numerosas recomendaciones, por lo que se debe aguardar a que, en el futuro, el equipo decida implementarlas o tenerlas en cuenta al concebir nuevas funcionalidades. Es importante recalcar que el equipo de DashAI tiene la potestad de decidir si merecen ser implementadas o no. Siempre está presente esa decisión que determina si la funcionalidad solicitada podrá ser procesada o será descartada, como sucedió con algunas recomendaciones que no estaban alineadas con los objetivos de la aplicación.

En cuanto al alcance y las limitaciones de la propuesta, se reconoce que el enfoque adoptado para abordar el problema de la falta de pruebas por parte de los usuarios ha permitido avanzar significativamente en la obtención de retroalimentación sobre la experiencia usuaria en la aplicación. Sin embargo, es importante señalar que, debido a la naturaleza dinámica del proyecto, algunas áreas pueden no haber sido abordadas completamente o pueden requerir una mayor exploración en futuras iteraciones. Por ejemplo, aunque se hayan realizado pruebas en ciertos aspectos de la aplicación, aún quedan aspectos por evaluar, como la accesibilidad para usuarios con discapacidades o la eficacia de la interfaz en diferentes dispositivos y entornos.

Sobre la validez de los resultados, se puede discutir que estos están respaldados por la participación de los usuarios y la aplicación de técnicas de evaluación reconocidas en el campo de UX. Sin embargo, es importante tener en cuenta que la interpretación de los resultados puede estar sujeta a sesgos individuales y a limitaciones inherentes al método utilizado. Por ejemplo, las preferencias personales de los usuarios o las condiciones específicas en las que se realizaron las pruebas pueden influir en la percepción de la usabilidad de la aplicación. Por lo tanto, es fundamental ser consciente de estas limitaciones y tomarlas en consideración al interpretar los hallazgos de las pruebas.

Algunos diseños en los métodos podrían haber sido diferentes para evitar sesgos o asegurar cumplimiento adecuado. Por ejemplo, la utilización de heurísticas en las pruebas exploratorias, realizadas por expertos en usabilidad en lugar de estudiantes que hayan cursado materias sobre interfaces de usuario, podría conducir a hallazgos significativamente distintos. Otro punto por mencionar es la adaptación del SUS, donde se realizaron cambios significativos en algunas preguntas. Estas modificaciones podrían haber limitado las preguntas a aspectos específicos y condicionado las respuestas de los usuarios de manera no óptima.

A pesar de la utilidad de los métodos empleados en el proyecto, también es necesario evaluar la aplicabilidad de las técnicas descartadas en ciertos contextos. Por ejemplo, la ausencia de técnicas como Pruebas A/B y *Card Sorting* limita la capacidad del equipo para realizar comparaciones directas entre diferentes diseños de interfaz y para estructurar eficientemente la información, respectivamente. Reconocer estas limitaciones es esencial para mejorar la selección de métodos en futuros proyectos, asegurando que se elijan los más adecuados según las necesidades y el contexto del desarrollo.

5.1 Trabajo a corto plazo

Entre los objetivos a corto plazo tras la implementación del prototipo se encuentra la ejecución de una prueba de usabilidad junto a varios integrantes de DashAI. Esta prueba permitirá evaluar diversos aspectos del prototipo de manera más cuantitativa, ya que involucrará a un mayor número de participantes.

La prueba de usabilidad buscará evaluar aspectos específicos como:

- Gráficas y tablas
- Optimización del modelo
- Explicabilidad
- *Plugins*

5.2 Trabajo futuro

Dado que el proyecto está concebido dentro de un marco temporal definido, inevitablemente quedará trabajo pendiente que deberá ser asumido por otros actores. Para asegurar la continuidad y el éxito del desarrollo de DashAI, se identifican varias tareas cruciales que requieren atención en futuras etapas del proyecto:

- Es fundamental corregir los problemas detectados durante la prueba de usabilidad de la versión prototipo. Estos problemas, identificados a través de la interacción directa con los usuarios, proporcionan un diagnóstico preciso de las áreas que necesitan mejoras inmediatas. Abordar estos problemas es esencial para garantizar que la interfaz y la funcionalidad de la aplicación sean robustas y libres de obstáculos que puedan frustrar la experiencia del usuario.
- La implementación de las recomendaciones ofrecidas que el equipo considere conveniente es otro paso crítico. No todas las sugerencias podrán ser implementadas de inmediato, por lo que será necesario un proceso de selección y priorización. Las recomendaciones de más impacto y viables deberán ser priorizadas para maximizar el beneficio para los usuarios y mejorar significativamente la usabilidad.
- Establecer un método centrado en el usuario para recopilar información sobre la usabilidad del sistema cuando se haga el *release* de la aplicación es vital. Este método debe ser diseñado para recoger datos cuantitativos y cualitativos de manera continua, permitiendo un monitoreo constante de la experiencia del usuario. La implementación de encuestas, análisis de métricas de uso y sesiones de retroalimentación post-lanzamiento contribuirán a identificar problemas emergentes y áreas de mejora continua.

5.3 Consejos para utilizar métodos UX

Cualquier persona que vaya a utilizar métodos centrados en usuarios debería considerar las siguientes recomendaciones, basadas en el desarrollo del proyecto y las conclusiones obtenidas:

- No temer a la práctica: Es esencial realizar pruebas continuamente y mantener una interacción constante con los usuarios. La práctica frecuente mejora la fluidez y la confianza en la conducción de pruebas.
- Realizar una prueba piloto: Siempre se debe llevar a cabo una prueba piloto con un usuario antes de la implementación formal. Esto ayuda a identificar errores no detectados previamente y permite que el facilitador se familiarice con las instrucciones y el flujo de la prueba.
- Mantener la neutralidad: Durante las interacciones con los usuarios, es importante no interferir en sus decisiones. Se debe permitir que los usuarios naveguen y tomen decisiones de manera autónoma para obtener una evaluación auténtica de la usabilidad, a menos que la prueba en sí lo requiera.
- Evaluar la pertinencia de las respuestas: Si los usuarios tienen dudas, se debe considerar si la respuesta influirá en la finalización de la tarea o si es simplemente una cuestión de curiosidad. Solo se deben responder las preguntas que no afecten la autenticidad de la prueba.
- Clarificar expectativas: Si los participantes preguntan "¿Qué hace esto?", se debe responder con "¿Qué esperas que haga?" o "¿Qué te imaginas que hace?". Esta técnica es útil para identificar si la interfaz es intuitiva para usuarios no familiarizados.
- Decidir el número de participantes: Es crucial determinar un número adecuado de participantes para las pruebas. Cinco personas suelen ser suficientes para identificar problemas de usabilidad si representan distintas características dentro del público objetivo. Sin embargo, para datos cuantitativos más robustos, se recomienda un grupo de más de 10 personas. Es importante que los participantes provengan de diversos contextos para evitar sesgos en la retroalimentación.

REFERENCIAS BIBLIOGRÁFICAS

[Barnum11] Barnum, C. M. (2011). *Usability Testing Essentials: Ready, Set ...Test!*. Burlington, Estados Unidos: Elsevier.

[Basusta15] Basusta, E. (2015). *Contextual Inquiry: Observing How Users Interact with Systems and Workflows*. User Testing. Recuperado de <https://www.usertesting.com/blog/contextual-inquiry>

[Drozdal20] Drozdal, J., Weisz, J., Wang, D., Dass, G., Yao, B., Zhao, C., Muller, M., Ju, L. y Su, H. (2020). *Trust in AutoML: Exploring Information Needs for Establishing Trust in Automated Machine Learning Systems*. *IUI '20: 25th International Conference on Intelligent User Interfaces*, 297-307.

[Endmann16] Endmann, A. y Keßner, D. (2016). *User Journey Mapping – A Method in User Experience Design*. *i-com*, 15, 105-110.

[Geisen17] Geisen, E. y Bergstrom, J. R. (2017). *Usability Testing for Survey Research*. Cambridge, Estados Unidos: Elsevier.

[Gerón19] Gerón, A. (2019). *Hands-on machine learning with Scikit-Learn and TensorFlow concepts, tools, and techniques to build intelligent systems*. Sebastopol, Estados Unidos: O'Reilly Media, Inc.

[Grigorescu19] Grigorescu, S., Trasnea, B., Cocias, T. y Macesanu, G. (2019). *A Survey of Deep Learning Techniques for Autonomous Driving*. *Journal of Field Robotics*, 37, 362-386.

[Hassenzahl08] Hassenzahl, M. (2008). *User Experience (UX): Towards an experiential perspective on product quality*. *IHM'08: Proceedings of the 20th Conference on L'Interaction Homme-Machine*, 11-15.

[Hooley20] Hooley, G., Nicoulaud, B., Rudd, J. M. y Lee, N. (2020). *Marketing Strategy and Competitive Positioning*. Harlow, Reino Unido: Pearson Education Limited.

[Kaplan23] Kaplan, K. (2023). *User Journeys vs. User Flows*. Nielsen Norman Group. Recuperado de <https://www.nngroup.com/articles/user-journeys-vs-user-flows/>

[Kumar12] Kumar, S. y Qadeer, M. A. (2012). *Application of AI in Home Automation*. *International journal of engineering and technology*, 4, 803-807.

[Lewrick20] Lewrick, M., Link, P. y Leifer, L. (2020). *The Design Thinking Toolbox: A Guide to Mastering the Most Popular and Valuable Innovation Methods*. New Jersey, Estados Unidos: Wiley.

[Lewis12] Lewis, J. R. (2012). *Usability Testing. Handbook of human factors and ergonomics*, 4, 1267-1312.

[Litjens17] Litjens, G. J., Kooi, T., Bejnordi, B. E., Setio, A. A., Ciompi, F., Ghafoorian, M., Laak, J. V., Ginneken, B. V. y Sánchez, C. I. (2017). *A survey on deep learning in medical image analysis. Medical image analysis*, 42, 60-88.

[Plathottam23] Plathottam, S. J., Rzonca, A., Lakhnori, R. y Iloeje, C. O. (2023). *A review of artificial intelligence applications in manufacturing operations*. Estados Unidos: *Journal of Advanced Manufacturing and Processing*. Recuperado de <https://aiche.onlinelibrary.wiley.com/doi/full/10.1002/amp2.10159>

[Rouhiainen18] Rouhiainen, J. (2018). *Inteligencia Artificial: 101 cosas que debes saber hoy sobre nuestro futuro*. Barcelona, España: Planeta.

[Salazar20] Salazar, K. (2020). *Contextual Inquiry: Inspire Design by Observing and Interviewing Users in Their Context*. Nielsen Norman Group. Recuperado de <https://www.nngroup.com/articles/contextual-inquiry/>

[Sachedina17] Sachedina, F. (2017). *What is Exploratory Testing?*. Global App Testing. Recuperado de <https://www.globalapptesting.com/blog/what-is-exploratory-testing>

[Wang19] Wang, D., Weisz, J. D., Muller, M., Ram, P., Geyer, W., Dugan, C., Tausczik, Y., Samulowitz, H. y Gray, A. (2019). *Human-AI Collaboration in Data Science: Exploring Data Scientists' Perceptions of Automated AI. Proceedings of the ACM on Human-Computer Interaction*, 3, 1-24.

[Wang21] Wang, D., Liao, Q. V., Zhang, Y., Khurana, U., Samulowitz, H., Park, S., Muller, M. y Amini, L. (2021). *How Much Automation Does a Data Scientist Want?*. Estados Unidos: ArXiv. Recuperado de <https://arxiv.org/abs/2101.03970>

ANEXOS

A.1. INVITACIÓN Y CONSENTIMIENTO INFORMADO

INVITACIÓN PARA PARTICIPAR EN PRUEBA DE USABILIDAD

El propósito del presente documento es invitarlos a participar en la prueba de usabilidad de la aplicación DashAI. Una aplicación AutoML enfocada en realizar predicciones mediante una interfaz gráfica, la cual será probada con usuarios en las dependencias de la UTFSM.

Para que usted pueda tomar una decisión informada, le explicaremos cuáles serán los procedimientos involucrados en la ejecución de la prueba, así como en qué consistiría su colaboración.

1. El estudio busca evaluar y probar DashAI con técnicas de diseño centrado en usuarios que permita medir de alguna manera la empatía y usabilidad de la aplicación.
2. Como herramientas de estudio se utilizarán entrevistas y/o formularios.
3. Los resultados de la prueba de usabilidad y las respuestas de la entrevista y/o formulario serán usados en el estudio directamente y existirá una reserva total de su identidad y la información que entregue. Su nombre no será revelado, y solo tendrán acceso a los datos recolectados el conjunto de investigadores del laboratorio y el profesor guía.
4. La entrevista se realizará de manera presencial de acuerdo con la disponibilidad del participante y de los investigadores.
5. Su participación consistirá en interactuar con una plataforma web, para ello seguirá instrucciones de un facilitador (investigador), y posteriormente en responder preguntas sobre su experiencia en el proceso, esto requerirá parte de su tiempo y conocimiento de la aplicación adquirido durante la prueba de usabilidad. Estimamos que el tiempo requerido para todo el proceso será de 30 minutos.
6. Los resultados y hallazgos de esta prueba serán utilizados para redactar un informe, el cual será anexado como parte de un trabajo de título una vez se dé por finalizado el proceso.
7. No se prevén riesgos asociados a su participación. Cualquier potencial pérdida de confidencialidad de los datos en este estudio será minimizada al almacenarlos en una ubicación segura.

8. Como participante, usted tiene derecho de manifestar sus dudas al investigador <daniel.roco@sansano.usm.cl> o profesora <claudia.lopez@usm.cl> en cualquier momento.
9. Tiene el derecho de retirarse del estudio en el momento en que usted lo considere necesario, sin que esto le traiga consecuencias.
10. La entrevista tiene motivos netamente estudiantiles e investigativos, es por eso por lo que nos adjudicamos el derecho de poder utilizar el material para próximos proyectos y/o exhibiciones a futuros. Solo con el fin educativo y de investigación.

EN CASO DE ACEPTAR LA INVITACIÓN PARA PARTICIPAR EN LA PRUEBA DE USABILIDAD DEBE LLENAR EL SIGUIENTE CONSENTIMIENTO INFORMADO.

CONSENTIMIENTO INFORMADO

Fecha _____

Mi nombre es _____, edad _____, trabajo en la empresa/organización _____, y me desempeño como _____.

Manifiesto mi voluntad de participar en este proyecto “DashAI”, por consecuencia acepto y autorizo que se me grabe y utilice mi imagen, declarando que Daniel Roco será el dueño exclusivo del material y podrá producirlo, reproducirlos, editarlo, adaptarlo y ejecutarlo públicamente.

Acepto los términos y condiciones para participar en la actividad a la que me han invitado, indicando que:

1. He leído lo anterior, o me lo han leído, y he entendido toda la información.
2. Cuando no entendí algo, pude preguntar, y han contestado a todas mis preguntas.
3. Sé que puedo decidir no participar, y no habrá consecuencia alguna. Si tengo alguna duda en cualquier momento de la actividad, puedo preguntar todas las veces que necesite.
4. Sé que puedo elegir participar, pero después puedo cambiar de opinión en cualquier momento sin consecuencias.
5. Sé que la información que entregue en esta actividad sólo la sabrán los integrantes de LabUX y el entrevistador. Si mis respuestas llegasen a ser publicadas, no

estarían relacionadas con mi nombre, así que nadie sabrá cuales fueron mi decisiones o respuestas.

6. De tener alguna pregunta sobre la actividad, después podré llamar o escribir al investigador encargado, que podrá responder todas mis preguntas y comentarios. Puedo contactarlos a través del correo <daniel.roco@sansano.usm.cl>
7. Si acepto participar en la actividad debo firmar el consentimiento y me entregarán una copia si tengo cualquier duda después.

Firma del Participante

Firma del Facilitador o
Investigador a cargo.

A.2. PRUEBA DE USABILIDAD VERSIÓN VERTICAL

Introducción

DashAI es una aplicación que está siendo desarrollada por el CENIA y se encuentra en fase de pruebas, cuya finalidad es realizar predicciones, mediante AutoML, sobre un *dataset* entregado por el usuario. Actualmente hay diseñada una interfaz que será presentada a continuación, sobre la cual se busca realizar una modificación centrada en la experiencia obtenida de los *testing* de usuarios, tal como el que nos encontramos ahora. En esta instancia vas a tomar el rol de “usuario de DashAI”, cuyo objetivo es seguir las tareas dadas y criticar objetivamente la aplicación.

Se entregará un consentimiento, que deberá ser completado y firmado por ti en caso de que quieras participar de esta experiencia. Si deseas no participar de esta experiencia, no hay ningún problema. [Entrega consentimiento y lee las cláusulas].

También, te recuerdo que, si deseas abandonar la experiencia en algún momento, no hay ningún problema.

El fin de esta experiencia es poner a prueba el avance actual de la aplicación mediante el desarrollo de las tareas con la mínima interacción del supervisor. No hay respuestas o acciones incorrectas, puedes preguntar todo lo que estime conveniente que consideres que no haya quedado claro.

Entrevista inicial

Me gustaría saber un poco más de ti.

- ¿En qué ciudad vives?

- ¿A qué te dedicas?
- ¿Cuáles son sus *hobbies*?
- ¿Qué edad tienes?
- ¿Cuánto tiempo llevas en el área de inteligencia artificial?
- ¿Estas participando en un proyecto actualmente?
- ¿Qué lenguaje de programación es el que más utilizas para *machine learning*?
- ¿Utilizaste AutoML alguna vez?
- ¿Cuál es el formato de datos (input) que más utilizas dentro del archivo (datos binarios, texto, números, etc) y la extensión del archivo que más usas?
- ¿Cuáles son los parámetros generales que más modificas en un modelo?

Escenarios de usuario

[Véase Tabla 1].

Entrevista final y agradecimientos

Primero que todo, agradecer tu tiempo por asistir aquí para realizar la prueba.

Me gustaría saber tu opinión acerca de esta experiencia mediante las siguientes preguntas.

1. ¿Cree que es adecuado el movimiento vertical del paso a paso a medida que se avanza en la interfaz?
2. ¿Los elementos visuales como tipo de letra, tamaño de letra, colores, etc, son adecuados?
3. En el caso de que un modelo se demore mucho en cargar, ¿qué te gustaría hacer con el procedimiento?
4. ¿Qué parámetros le gustaría poder modificar y que no aparecen en la interfaz?
5. ¿Qué información le gustaría que se muestre al finalizar la ejecución que no aparecen actualmente?
6. ¿Cree que sea necesario que se muestre detalladamente cómo el modelo trabajó los datos?
7. Dado que en algunos casos se trabaja con datos sensibles, ¿preferiría una aplicación personal de DashAI en un ordenador o una aplicación web?

8. ¿Tienes algún comentario o crítica sobre la interfaz?

9. ¿Usaría usted esta aplicación?

Gracias por venir.

A.3. COMENTARIOS Y OBSERVACIONES

Usuario/Tarea	T1	T2	T3	T4
Usuario 1	El usuario necesitó visualizar el archivo para entender lo que tenía que hacer.	El usuario comentó que no sabía si al elegir el modelo ya venían incluidos los parámetros predeterminados.	El usuario no tardó mucho en realizar la predicción porque pudo visualizar los archivos durante la primera tarea.	El usuario experimentó una demora significativa en la carga de los resultados, por lo que expresó su deseo de contar con una opción para cancelar un experimento en caso de necesidad.
Usuario 2	El usuario experimentó dificultades para encontrar el archivo que necesitaba utilizar, además de expresar su insatisfacción con el tiempo de espera necesario para subir un archivo.	-	Al usuario le incomodó tener que consultar el archivo para determinar qué dato debía insertar.	Tuvo que reiniciar la aplicación, dado que los modelos no se limpiaron una vez se apretó el botón respectivo. El usuario notó que solo se presentan los resultados del primer modelo entrenado y expresó su deseo de verlos separadamente o, como mínimo, que se muestre el promedio de los resultados.
Usuario 3	-	El usuario realizó predicciones con dos modelos, pero quedó confuso al darse cuenta de que solo se le entregó el resultado de uno.	Al usuario le incomodó tener que consultar el archivo para determinar qué dato debía insertar.	Hubo un error al intentar predecir el modelo modificado, lo que resultó en la necesidad de reiniciar la aplicación.
Usuario 4	-	-	Al usuario le incomodó tener que consultar el archivo para determinar qué dato debía insertar.	El usuario no tuvo ningún problema para comprender las modificaciones que debían realizarse.

A.4. RESPUESTAS DE LA ENTREVISTA FINAL

Usuario/Pregunta	¿Cree que es adecuado el movimiento vertical del paso a paso a medida que se avanza en la interfaz?	¿Los elementos visuales como tipo de letra, tamaño de letra, colores, etc, son adecuados?	En el caso de que un modelo se demore mucho en cargar, ¿que te gustaría hacer con el procedimiento?	¿Qué parámetros le gustaría poder modificar y que no aparecen en la interfaz?	¿Qué información le gustaría que se muestre al finalizar la ejecución que no aparecen actualmente?	¿Cree que sea necesario que se muestre detalladamente cómo el modelo trabajó los datos?	Dado que en algunos casos se trabaja con datos sensibles, ¿preferiría una aplicación personal de DashAI en un ordenador o una aplicación web?	¿Tienes algún comentario o crítica sobre la interfaz?	¿Usaría usted esta aplicación?
Usuario 1	No le desagrada la disposición, pero está más acostumbrado a ver la información en formato horizontal.	Le gusta la presentación en modo oscuro y prefiere las interfaces que ofrecen esta opción. Considera que las que no cuentan con este modo se perciben como obsoletas.	El usuario sugiere que le gustaría ver los datos entrenados hasta ese momento, incluyendo cuántos datos han sido entrenados.	Sugiere incluir una breve descripción de cada parámetro, ya que suele consultar la documentación cada vez que utiliza modelos. Además, sugiere establecer límites claros sobre el rango de valores aceptables para cada parámetro.	El usuario sugiere incluir gráficos de curvas para los modelos entrenados y desea saber si se separan en conjunto de pruebas y de validación. En caso de que los datos sean escasos, le gustaría que se indique cuántos son. Además, solicita que se explique qué representan cada porcentaje mostrado y que se muestren cuáles datos fueron clasificados erróneamente.	Le gustaría, pero que sea opcional, ya que no lo ve como una utilidad significativa para alguien que no esté familiarizado con el tema.	Aplicación Web.	Al usuario le agrada la simplicidad del enfoque. Sugiere que al realizar la carga de datos, se incluya una porción de los datos, por ejemplo, unas cinco filas, con el fin de visualizar qué tipo de datos contiene el conjunto de datos.	Si.
Usuario 2	Lo encuentra bien, pero el espacio vacío que queda debajo de la carga de datos no le resulta satisfactorio.	El usuario prefiere el modo oscuro y sugiere no implementar un modo claro.	El usuario desea que se muestre el progreso durante el proceso para determinar si es conveniente detenerlo o dejarlo continuar. También sugiere que se entregue un modelo con la situación actual en caso de interrupción.	El usuario desea la capacidad de cargar un modelo propio. Además, sugiere que exista una configuración simple y avanzada para ajustar los parámetros de los modelos.	El usuario sugiere que la aplicación muestre cuál es el modelo más utilizado por los usuarios y también que incluya una matriz de confusión para evaluar el rendimiento del modelo.	Explicar qué significan los resultados.	Aplicación de ordenador.	El usuario prefiere que cada sección de la aplicación tenga una identificación numérica o textual para facilitar la navegación, en lugar de depender únicamente del desplazamiento vertical.	Si.
Usuario 3	Se acostumbra, pero prefiere el horizontal o simplemente que cambie de vista.	Si.	Detenerlo o entrenar otro al mismo tiempo.	Tener una configuración avanzada sobre todos los modelos.	El usuario sugiere incluir gráficos para visualizar el rendimiento de los modelos, así como también mostrar la cantidad de datos clasificados erróneamente.	Mostrar en cómo afecto el rendimiento el haber modificado cierto parámetro.	Aplicación de ordenador.	Que haya un enlace a la documentación de Sklearn para investigar ciertos elementos. Que se validen los errores antes de continuar hacia otro componente.	Si.
Usuario 4	No le desagrada la disposición, pero está más acostumbrado a ver la información en formato horizontal.	Si.	Poder entrenar otro modelo en paralelo y parar en caso de que se demore mucho.	Sugiere que exista una explicación de para qué sirve cada parámetro.	Una nueva sección para poder comparar los distintos modelos.	El usuario desea acceder a la información detallada sobre cómo se entrenaron los datos.	Aplicación Web.	El botón para modificar los parámetros avanzados no es lo suficientemente visible.	Si.

B.1. USER JOURNEYS

Cargar dataset tabular	1) Entrar en la aplicación y crear un nuevo experimento. 2) Seleccionar la task adecuada. 3) Seleccionar formato del archivo. 4) Configurar parámetros para el dataset . 5) Cargar el archivo.	Ejecutar un experimento	1) Configurar un experimento si aún no ha sido configurado. 2) Ejecutar el experimento. 3) Visualizar que el experimento se está ejecutando (barra de progreso, contador e tiempo, etc). 4) Visualizar el feedback de la aplicación sobre el modelo que se está entrenando actualmente. 5) Esperar hasta que el experimento ha terminado su ejecución.	Configurar un experimento	1) Seguir los pasos de "Cargar dataset tabular". 2) Abrir la configuración de parámetros de uno de los modelos que se añadió antes. 3) Interactuar con el formulario modificando los parámetros del modelo. 4) En caso de necesitar ayuda con el formulario, utilizar las ayudas que incluyen descripciones y enlaces a la documentación oficial sobre algún parámetro específico. 5) En caso de introducir valores incorrectos, se avisa por el formulario que debe arreglar estos valores. 6) Cuando esté listo, guardar la configuración de parámetros para el modelo.
Visualizar resultados de un experimento	1) Seleccionar un modelo específico. 2) Visualizar una tabla con varias métricas para este modelo, además de la métrica principal (e.g precision, recall , accuracy , para clasificación numérica) 3) Visualizar los parámetros con los que se entrenó a este modelo. 4) Visualizar gráficos relevantes para este modelo (e.g matriz de confusión, curva ROC en clasificación numérica). 5) Opcionalmente repetir los pasos de esta fase para otro modelo. 6) Tras lo anterior, volver al resumen comparativo de los resultados del experimento. 7) Seleccionar dos modelos para compararlos. 8) Visualizar métricas específicas y gráficos que comparan los resultados de estos dos modelos.	Predecir con un modelo	1) Seleccionar un modelo entrenado. 2) Completar un formulario para entregar un input al modelo. 3) Enviar input al modelo y esperar por la predicción. 4) Visualizar la predicción del modelo.	Usar resultados de experimentos anteriores	1) Realizar los pasos de "Visualizar resultados de experimento". 2) Seleccionar uno de los modelos e intentar modificar su configuración de parámetros. 3) Tomar un atajo que permita ir directamente a la configuración de parámetros del modelo. 4) Desde la configuración del modelo seleccionado anteriormente, crear una copia de este para modificar sus parámetros. 5) Configurar algunos parámetros del nuevo modelo. 6) Ejecutar nuevamente el experimento seleccionado anteriormente y el nuevo modelo (copia). 7) Realizar los pasos de "Visualizar resultados de experimento" enfocándose en comparar el nuevo modelo con el que se tomó como base.
Cerrar la interfaz mientras se entrenan modelos	1) Realizar los pasos de "Cargar dataset tabular". 2) Realizar los pasos de "Configurar un experimento". 3) Ejecutar un experimento. 4) Mientras se encuentra en la pantalla de carga, cerrar la interfaz gráfica.	Re-abrir la interfaz luego de entrenar modelos	1) Entrar en la interfaz y visualizar resumen de los experimentos. 2) Dirigirse a la tabla de resultados. 3) Realizar los pasos de "Visualizar resultado de experimento" para un modelo entrenado.		

B.2. RESULTADOS DE LA COMPARACIÓN

User Journey/Software	MLFlow	Weights & Biases	WEKA	RapidMiner Studio (Educational)	Perceptilabs	Akkio	H2O Flow	KNIME	Hugging Face Auto Train	Modulos AI
Cargar dataset tabular	Desde consola	Desde consola	Al ingresar a la interfaz hay que seleccionar el botón "Explore" y seleccionar "Open"	A la derecha hay un botón para importar un archivo, en el cual aparece un Wizard que guía al usuario en el Preprocesado	En "Overview" está la opción de cargar un modelo popular de la red o uno propio. Luego aparece una visualización del preprocesado donde se puede elegir el porcentaje de Training, Validation y Test.	Permite subir archivos localmente o desde algun servidor online.	En la opción "Data/Import Files" se pueden subir los archivos y parsearlos. En "Data-Split Frame" es posible dividir en Test y Validation	Se deben crear nodos (es necesario ver un tutorial) y cargar el dataset a tal elemento. Se puede trabajar el Preprocesado, pero no los datos referentes a Training y Validation.	Existe un menú que aparece luego de elegir el tipo de modelo a ocupar, en donde se puede subir un archivo o elegir uno de Hugging Face Hub	Se suben los archivos a través de la interfaz. Una vez se suben, el sistema comienza a validar los datos.
Configurar un experimento	Desde consola	Desde consola	Cargado el archivo, en las opciones superiores se puede configurar el Clasificador y atributos de entrenamiento	Se debe establecer un bloque y seleccionar uno de los pocos tipos de modelos disponibles, en donde es posible configurarlo	Seleccionado el archivo, aparece una vista detallada del modelo donde permite configurar Epochs, batch size, Loss, Learning rate, Optimizer, beta1 y Beta2	Permite trabajar el Preprocesado de los datos. Practicamente no hay configuración	Permite seleccionar el modelo a ocupar, así como una gran cantidad de configuraciones para cada una de ellas	Se crea un nodo que permita entrenar los datos mediante unos modelos predeterminados	Permite una configuración básica de los parámetros a elegir. Para cada tipo de evaluación, existen distintos modelos pre establecidos a ocupar, los cuales se pueden elegir manualmente o decidir al sistema. Luego de cargar un archivo, se seleccionan los parámetros sobre valores objetivos y porcentajes del Split de datos.	Permite realizar una configuración de cual es el parámetro a evaluar, además de extractores y una serie amplia de modelos a utilizar
Ejecutar un experimento	Desde consola	Desde consola	Luego de configurar se debe realizar la prueba en la ventana "Classify"	Arriba de la interfaz hay una flecha que permite ejecutar el modelo y aparece una vista donde muestra que el modelo se esta ejecutando (no hay barra progresiva)	Al seleccionar la configuración se abre el botón iniciar y aparece una barra indicando el estado actual del entrenamiento, junto a muchas métricas.	Acceder a la sección de "Predict" y permite seleccionar el dataset a utilizar, configurando la calidad esperada (a mejor calidad, mayor tiempo de espera)	Existe una barra que indica el estado actual del entrenamiento	Click derecho en un nodo y apretar "Execute". En la consola de abajo aparece el progreso actual de cada nodo	Luego de seleccionar las configuraciones, se puede iniciar con un botón en la parte superior.	Luego de seleccionar la configuración, existe abajo a la derecha un botón para iniciar el proceso siguiente
Visualizar resultados de un experimento	Aparecen los resultados de manera detallada con gráficos y métricas adecuadas	Aparecen los resultados de manera detallada con gráficos y métricas adecuadas	En la propia ventana de la configuración se ve el resultado, al igual que en una vista de "Visualizar"	Diferentes tipos de visualizaciones de métricos y gráficos, para el modelo en general y cada nodo	Gráficos con detalle de los diferentes parámetros. No hay comparación entre modelos.	Información detallada para cada campo evaluado en la predicción, tanto con métricas, gráficos y una interfaz interactiva para variar los parámetros	Detalle con métricas y gráficos, junto con las variables más importantes. Las métricas son muy detalladas.	Una vez terminado de ejecutarse un nodo se debe seleccionar con click derecho y usar la opción de mostrar resultados	Se entrenan varios modelos al mismo tiempo, de modo que luego podemos visualizar los resultados de cualquiera de ellos con una vista que se despliega lateralmente	Visualización en una tabla donde están todos los modelos entrenados, mostrando el tamaño, el tiempo de duración y la posibilidad de generar un Dashboard que especifica más en detalle el resultado obtenido. Cabe destacar que se puede descargar un archivo en donde se muestran ejemplo de datos testeados y los resultados obtenidos con el modelo.
Predicir con un modelo	No	No	No	No	No	No	Se pueden realizar predicciones, pero el usuario no puede decidir los inputs	No	No	No existe en sí una sección para predecir, pero sí se puede re-entrenar un modelo, eligiendo si se quiere aumentar la precisión o el desempeño general.
Usar resultados de experimentos anteriores	Se pueden observar los experimentos realizados y compararlos entre sí	Se pueden observar los experimentos realizados y compararlos entre sí	Visualización de experimentos pasados, siempre y cuando no se haya cerrado la aplicación	Archivos se quedan guardados en equipo	En la sección "Overview" hay una lista con todos los experimentos realizados y con acceso a ellos.	Existe una vista en donde estan todos los modelos entrenados.		Los experimentos quedan guardados en el área de trabajo definido al inicio, con lo cual quedan almacenados en disco y se puede visualizar en cualquier momento	Existe una vista en donde estan todos los modelos entrenados	Los experimentos quedan guardados en el área de trabajo definido al inicio, a los cuales se puede acceder fácilmente desde el menu lateral.
Cerrar interfaz mientras se entrenan modelos	Si, la interfaz gráfica es solo para visualizar gráficos	Si, la interfaz gráfica es solo para visualizar gráficos	Minimizando aplicación (no en segundo plano)	No	Con lo que se demora en iniciar el programa no dan ganas de pasarlo a segundo plano por miedo a que se cierre.	Pestaña de navegador (si se cierra termina)	Si se cierra la interfaz se cancela el modelo	Se puede minimizar	Se puede cerrar la pestaña del navegador y se sigue ejecutando el modelo	En la demo de video no aparece
Re-abrir interfaz luego de entrenar modelos	Si, la interfaz gráfica es solo para visualizar gráficos	Si, la interfaz gráfica es solo para visualizar gráficos	No	Archivos se quedan guardados en equipo	Se quedan guardados los modelos entrenados	Se quedan guardados los modelos entrenados		Se puede siempre y cuando se haya especificado el área de trabajo anterior y haber guardado el trabajo	Se quedan guardados los modelos entrenados	Se quedan guardados los modelos entrenados
Interfaz	Agradable, con simplicidad en su representación. Las acciones para poder iniciar un modelo son desde consola.	Simple y con excelente representación de gráficos, pero las acciones para iniciar un modelo son desde consola.	Interfaz antigua y requiere ser instalada, la cual no tiene para iniciar desde un localhost. Muchas de sus opciones son difíciles y de seguir y no hay unas instrucciones claras del flujo de acción	Poco intuitiva y muy invasiva. Visualmente se ve compleja cuando se diseña un modelo un poco más complejo.	Tema oscuro, buena separación de secciones, colores adecuados, complejidad en la solución, intuitivo y con muchas opciones de personalización.	Aplicación intuitiva con buen sistema paso a paso, además de que es totalmente web, pero prácticamente no posee configuración sobre el modelo.	Interfaz muy pequeña, los botones cuesta encontrarlos, la distribución de su información se centra mucho en la vista principal, dejando las demás secciones muy poco trabajadas y casi sin funciones útiles.	Interfaz muy grande que a primera vista no es agradable. Para cada acción, (preprocesado, entrenamiento, visualización, entre otros) se requiere la creación de nodos, los cuales requieren conectarse entre sí, complicando la visualización final.	Cuando se esta formateando la data, la información de estado actual no es clara, en donde no se sabe el estado real del proceso (ya sea si se sube un archivo o se utiliza uno de internet)	Interfaz agradable a la vista, con un buen menú lateral para acceder a las partes más importantes. Además, la función "Improve model" me parecio muy buena idea, ya que sirve para re-entrenar un modelo ya usado, sin la necesidad de crear uno nuevo
Facilidad instalacion	Se complica un poco la instalación, pero solo requiere tiempo, no tantos conocimientos.	Instalación fácil y rápida.	Instalación fácil y rápida.	Instalación fácil.	Muy difícil su instalación, además de que pueden aparecer errores durante esta, los cuales debe resolver el usuario y requiere de mucho tiempo.	No requiere instalación	Instalación fácil y rápida.	Instalación fácil y rápida.	No requiere instalación	No se pudo comprobar
Solicitud licencia	Gratis	Gratis	Gratis	Requiere licencia	Gratis	Gratis (acciones limitadas)	Gratis	Gratis	Gratis	No se pudo comprobar

C.1. OBSERVACIONES OBTENIDAS

Procedimiento/ Actividad	Carga y exploración de datos	Selección del train / test split	Valores faltantes e imputación	Estandarización	Creación de la red neuronal	Modificación de la red neuronal
Procedimientos observados	Cada estudiante comenzó importando las bibliotecas necesarias y cargando los datos en un dataframe de pandas, asegurándose de que los datos se cargaran correctamente, revisando errores y verificando la presencia de todas las columnas y filas. Tras esto, realizaron una revisión visual inicial utilizando métodos como df.head() y df.info(), lo que permitió obtener una visión general de la estructura de los datos, tipos de columnas y la presencia de valores nulos o atípicos. Cada estudiante tomó notas o realizó comentarios sobre sus observaciones, aunque el nivel de detalle varió.	Cada estudiante comenzó identificando y separando las variables independientes (features) de la variable objetivo (target). Esta etapa fue crucial para asegurar que el modelo se entrenara con los datos correctos. Los estudiantes aplicaron la función train_test_split para dividir el conjunto de datos, siguiendo en su mayoría la proporción sugerida de 70% para entrenamiento, 20% para validación y 10% para prueba. Algunos exploraron diferentes proporciones, reflexionando sobre cómo esto podría afectar el rendimiento del modelo. Después de realizar la división, los estudiantes verificaron el tamaño de cada conjunto para confirmar que la separación se realizó correctamente. Algunos experimentaron con diferentes valores para el random_state en un esfuerzo por comprender su impacto en la división de los datos.	Inicialmente, los estudiantes identificaron los valores que necesitaban ser tratados como faltantes y los reemplazaron con NaN usando el método replace. Posteriormente, realizaron un conteo de los valores faltantes en cada columna y visualizaron estos conteos mediante un gráfico de barras. Esta etapa les permitió comprender la magnitud del problema de los datos faltantes en el conjunto de entrenamiento.	Los estudiantes crearon nuevos conjuntos de datos estandarizados para el conjunto de entrenamiento, modificando en la imputación, así como para los conjuntos de validación y prueba	La mayoría de los estudiantes construyó modelos utilizando una arquitectura de red neuronal con al menos una capa oculta. La elección de la cantidad de neuronas y la función de activación varió, aunque hubo una tendencia a utilizar ReLU para las capas ocultas. Los estudiantes comprendieron la importancia de seleccionar una función de pérdida adecuada para clasificación binaria, siendo la Entropía Cruzada Binaria una elección común. Sin embargo, la selección del optimizador y la tasa de aprendizaje mostró mayor variabilidad, reflejando diferentes niveles de comprensión sobre su impacto en el entrenamiento. Durante el entrenamiento de sus modelos en los conjuntos de datos preparados, los estudiantes prestaron atención a la necesidad de validar el rendimiento del modelo con datos separados. Se observaron diferencias en la cantidad de epochs utilizados para el entrenamiento y en cómo interpretaron los resultados de validación. Algunos estudiantes adoptaron un enfoque crítico al analizar el rendimiento de su modelo, identificando posibles casos de underfitting o overfitting y ajustando parámetros como la tasa de aprendizaje para mejorar los resultados.	Más allá de las inicializaciones estándar, algunos estudiantes probaron diferentes inicializadores de pesos para observar sus efectos en el aprendizaje y la estabilidad de la red. Esta exploración les permitió comprender cómo una correcta inicialización puede prevenir problemas como el vanishing gradient y mejorar la convergencia. Además del Dropout, los estudiantes implementaron otras técnicas para reducir el overfitting, como la regularización de los datos de entrada y la manipulación de la arquitectura de la red.
	La mayoría utilizó df.describe() para obtener estadísticas descriptivas de las columnas numéricas, lo cual fue crucial para entender la distribución de los datos, identificar posibles outliers y obtener una visión general de las características numéricas del conjunto de datos. Se observó una variedad en la aplicación de técnicas de visualización: algunos estudiantes optaron por visualizaciones básicas como gráficos de barras e histogramas, mientras que otros experimentaron con visualizaciones más complejas, como matrices de correlación y pairplots. La elección de las visualizaciones estuvo influenciada tanto por la familiaridad con las herramientas como por los objetivos específicos de la tarea.					

D.1. FORMATO UTILIZADO

Heurística de Nielsen	Problema Identificado	Identificación del Componente	Ejemplo/Evidencia	Severidad (Baja/Media/Alta)	Recomendaciones/Sugerencias de Mejora	Comentarios Generales
Visibilidad del estado del sistema						
Relación entre el sistema y el mundo real						
Control y libertad del usuario						
Consistencia y estándares						
Prevención de errores						
Reconocer en lugar de recordar						
Flexibilidad y eficiencia de uso						
Diseño estético y minimalista						
Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores						
Ayuda y documentación						

D.2. PROBLEMAS Y MEJORAS ENCONTRADOS

	Problema	Problema	Problema	Problema	Problema	Mejora	Problema	Problema	Mejora	Mejora
Evaluaciones / Hallazgos	Cuando se sube a la aplicación un archivo que tenga el mismo nombre que uno de los ya cargados, tira un error, pero no indica cuál es el error ni como solucionarlo.	Al ordenar las columnas de las tablas, si se aprieta "Hide Columns" y luego uno cierra esa ventana emergente de configuración, no hay opción de poder restaurar tales columnas.	No es posible ocultar ninguna columna, ya sea mediante "Hide Column" o con "Manage Columns" en el módulo de Experimentos.	Cuando se selecciona subir un archivo CSV o JSON, es posible cargar uno de otra extensión, y aparece el diálogo verde, indicando de que se cargó correctamente, lo cual no es así.	Una vez se ejecuta un experimento, el botón "Finished" no cierra la ventana actual, con lo que la única opción para cerrarlo es haciendo clic por fuera de ella.	Poder exportar un modelo generado, además de los resultados asociados. Con la finalidad de replicar el modelo o almacenarlo.	No es posible editar el "Splits Configuration" una vez se almacena un <i>dataset</i> , con lo cual la única opción que queda es volver a cargar el mismo archivo y cambiar la configuración.	Cuando uno quiere exportar los resultados de un experimento, si selecciona "Print", se obtiene una imagen blanca sin ningún dato.	Hacer que el botón de DashAI sea interactivo, y que al presionarlo uno sea redirigido hacia la página principal.	Poder seleccionar qué métricas serán visualizadas en los resultados, antes de que se ejecute el experimento.
Cantidad de usuarios que lo notificaron	2	3	1	2	5	1	1	2	4	1
Severidad promedio (Baja/Media/Alta)	Media	Baja	Baja	Baja	Baja	Media	Baja	Baja	Baja	Baja
Heurística asociada	Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores	Prevención de errores	Control y libertad del usuario	Prevención de errores	Consistencia y estándares	Flexibilidad y eficiencia de uso	Control y libertad del usuario	Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores	Consistencia y estándares	Flexibilidad y eficiencia de uso

E.1. PRUEBA DE USABILIDAD VERSIÓN HORIZONTAL

Introducción

<Nombre del usuario>, en primer lugar, quiero agradecerte por dedicar tu tiempo a participar en esta sesión de prueba de usabilidad para DashAI. Antes de iniciar, es importante brindarte un contexto sobre la aplicación que evaluarás. DashAI, desarrollada por el CENIA en conjunto con otras instituciones, se encuentra en una fase de pruebas con el propósito de realizar predicciones mediante AutoML a partir de *datasets* proporcionados por los usuarios. Actualmente, estamos trabajando en una interfaz que te

presentaremos a continuación, buscando implementar mejoras basadas en las experiencias recogidas de pruebas de usabilidad como esta.

Durante la sesión, te invitamos a completar un conjunto de cuatro actividades y una encuesta, estimando un tiempo total de aproximadamente 30 minutos para el proceso. Te pedimos que realices estas tareas de manera independiente, como si no estuviese aquí para guiarte. Es crucial entender que el objetivo de esta sesión es evaluar el diseño de la aplicación, no al usuario. Por lo tanto, no hay respuestas correctas o incorrectas y no debes preocuparte por el resultado. Lo más importante es que, mientras avanzas en las tareas, compartas en voz alta tus pensamientos y experiencias.

Cualquier aspecto que te resulte confuso, frustrante o que suscite dudas, te animamos a comunicarlo. Tus observaciones son extremadamente valiosas y contribuirán significativamente a la mejora del diseño de DashAI.

[Se solicita permiso para grabar la sesión].

Entrevista inicial

¿En qué ciudad vives?

¿En qué proyectos o tareas has aplicado *machine learning*?

¿Has usado herramientas AutoML antes? Si es así ¿cuál fue el motivo?

Contexto para realizar las tareas

Imagina que eres un científico de datos trabajando en un proyecto de investigación sobre la diabetes. Tienes un conjunto de datos que contiene la información de pacientes y te gustaría utilizar DashAI para analizar los datos y predecir los resultados de los tratamientos. Quieres comparar diferentes modelos para decidir cuál es el más adecuado para tu estudio.

Tareas

Tarea 1: Tienes un archivo `diabetes.csv` que contiene los datos médicos que necesitas cargar en DashAI, para lo cual buscas en la plataforma la opción de subir el archivo y añadirlo a la lista de *datasets*. Usando la interfaz, fíjate si puedes subir este archivo, recuerda pensar en voz alta y decirme cuando creas que la tarea ha sido finalizada.

Tarea 2: Ahora que tu *dataset* está cargado, quieres crear un experimento utilizando un modelo específico para hacer predicciones. Navega por la interfaz para encontrar dónde y cómo puedes crear un nuevo experimento y posteriormente ejecutarlo.

Tarea 3: Tu experimento ha finalizado y estás ansioso por ver los resultados detallados para entender el rendimiento del modelo que has utilizado. Localiza y revisa los resultados detallados de tu experimento en la plataforma.

Tarea 4: Para tomar la mejor decisión para tu proyecto de investigación, necesitas comparar el rendimiento de dos modelos diferentes. Crea un nuevo experimento para que incluya dos modelos diferentes y ejecútalos. Luego, busca cómo puedes comparar sus resultados dentro de DashAI.

Entrevista final y agradecimientos

Primero que todo, agradecer de nuevo tu tiempo por asistir aquí para realizar la prueba.

Me gustaría saber tu opinión acerca de esta experiencia mediante las siguientes preguntas.

1. En general, ¿cuál ha sido tu experiencia al realizar estas tareas propuestas en el sitio?
2. Durante el desarrollo de las tareas, ¿tuvo alguna dificultad? Si es así, ¿podría especificar cuáles fueron?
3. Respecto a la secuencia de ejecución de las tareas, ¿la encontró fácil de seguir o considera que faltan pasos explicativos?
4. Los íconos presentes en el sitio, ¿le parecen legibles, reconocibles y explicativos?
5. En términos de diseño visual, ¿cómo considera el sitio: simple, equilibrado o recargado de información?
6. ¿Considera que los elementos visuales, como imágenes, tablas y texto, están bien ubicados y contribuyen a la claridad del sitio?
7. Durante el uso del sitio, ¿identificó alguna funcionalidad o característica adicional que le hubiera gustado encontrar?
8. Por último, ¿tiene algún comentario o sugerencia específica sobre la interfaz evaluada?

Si tuvieras que calificar la aplicación, ¿Qué nota pondría del 1 al 10?

Gracias por venir.

E.2. COMENTARIOS Y OBSERVACIONES

Usuario/Tarea	T1	T2	T3	T4
Usuario 1	No estaba seguro si debía separar por comas el archivo subido. Sin embargo, le gustó la separación de los datos en conjuntos (<i>splits</i>).	Experimentó confusión después de crear y ejecutar un experimento debido a que no entendió qué hizo, ya que no había definido cuál era su variable objetivo. Además, el usuario nunca especificó sobre qué realizar la predicción. También considera que algunas explicaciones sobre los parámetros no son suficientemente detalladas.	Muestra una actitud confusa ya que no comprende cómo le aporta la información detallada asociada a "INFO".	Exploró otras formas de comparación mediante la adición de nuevas columnas. También expresó aprecio por la función de filtrado de elementos. Además, investigó el archivo CSV, ya que durante la carga nunca pudo visualizar qué datos contenía el archivo.
Usuario 2	Encontró útil la descripción de las tareas, ya que le permitió deducir fácilmente qué tarea tenía que realizar.	Le costó reconocer que era necesario colocar un nombre al experimento para poder continuar con el proceso.	-	No utilizó la tabla para comparar los modelos, lo hizo en base al detalle de cada uno.
Usuario 3	Se dirigió a la sección "Datasets" para subir los archivos basado en el nombre y su relación con los archivos. Al principio, no entendió qué debía hacer con las <i>Tasks</i> , ya que no había visto los datos con los que trabajar. Además, no está acostumbrado a utilizar el <i>split</i> de validación, ya que suele trabajar con datos nuevos para esa etapa.	El usuario no comprende por qué tiene que volver a seleccionar la tarea de clasificación tabular si ya la eligió durante la carga de datos.	Encuentra algunos parámetros en los detalles que no entiende completamente, pero le gustaría tener una breve descripción para comprenderlos mejor. Además, señala que nunca seleccionó la variable objetivo, por lo que no sabe a qué apuntan los resultados mostrados en el detalle.	-
Usuario 4	Asume que la separación de los datos es por coma, queda con dudas al respecto.	Comento que le gustó el diseño de la interfaz.	No comprende qué significa el "status 3" en los detalles y desea una explicación al respecto. Además, menciona que le gustaría ver el detalle de los elementos que fallaron en la predicción.	-
Usuario 5	Habría preferido que el nombre del archivo se hubiera completado automáticamente en lugar de tener que ingresarlo manualmente.	-	No comprende sobre qué se está realizando la predicción. Además, le gustaría que los parámetros con valores nulos estuvieran ocultos y habría preferido validar el modelo con nuevos datos.	Le hubiera gustado tener unos gráficos para comparar los resultados obtenidos.

E.3. RESPUESTAS DE LA ENTREVISTA FINAL

Usuario/Pregunta	En general, ¿cuál ha sido tu experiencia al realizar estas tareas propuestas en el sitio?	Durante el desarrollo de las tareas, ¿tuvo alguna dificultad? Si es así, ¿podría especificar cuáles fueron?	Respecto a la secuencia de ejecución de las tareas, ¿la encontró fácil de seguir o considera que faltan pasos explicativos?	Los iconos presentes en el sitio, ¿le parecen legibles, reconocibles y explicativos?	En términos de diseño visual, ¿cómo considera el sitio: simple, equilibrado o recargado de información?	¿Considera que los elementos visuales, como imágenes, tablas y texto, están bien ubicados y contribuyen a la claridad del sitio?	Durante el uso del sitio, ¿identificó alguna funcionalidad o característica adicional que le hubiera gustado encontrar?	Por último, ¿tiene algún comentario o sugerencia específica sobre la interfaz evaluada?	Si tuvieras que calificar la aplicación, ¿Qué nota pondría del 1 al 10?
Usuario 1	Las acciones son intuitivas de realizar, pero el usuario no tiene claro el objetivo para el cual está realizando la predicción.	Ninguna.	Encontró la interfaz fácil de seguir, pero echó de menos la capacidad de consultar los conjuntos de datos. Consideró que la separación entre Experimentos y Resultados era confusa y sugirió que sería más útil pasar directamente de un experimento a sus resultados sin tener que navegar a través de pestañas separadas.	Si.	Equilibrado tirado a simple.	Si.	Incluir una pestaña con ejemplos básicos para facilitar el entendimiento, mejorar la descripción de los parámetros y elementos, y sugiere que sería útil tener un enlace al modelo para principiantes.	No.	6
Usuario 2	Fácil e intuitivo, hay suficiente texto explicativo para describir cada opción a seleccionar, lo cual ayuda al usuario a no perderse.	Ninguna.	Fácil de seguir.	Si.	Equilibrado tirado a simple, minimalista pero funcional.	Si.	Sugiere añadir la opción de subir un modelo pre-entrenado en la sección de configuración de modelos en Experimentos, para facilitar su utilización.	Que el ícono de DashAI rediriga al menú principal.	7
Usuario 3	Buena experiencia.	Ninguna.	La transición desde la sección de Carga de Datos a la de Experimentos no quedó clara.	Si, salvo el botón <i>Update</i> presente en las tablas.	Simple, pero la sección de los resultados está un poco cargada.	Encontró un poco difícil visualizar el menú izquierdo de los resultados.	Posibilidad de explorar los datos que se cargarán y seleccionar la variable objetivo antes de proceder, para facilitar el proceso de configuración.	No.	8
Usuario 4	Muy intuitivo, es muy bueno el poder crear muchos modelos distintos al mismo tiempo.	Ninguna.	Muy intuitivo, le gustó que fuera un paso por pestaña (módulos)	Si.	Simple.	Si.	Falta incluir el modelo de redes neuronales, barra de progreso en la ejecución de los experimentos y poder cancelar entrenamientos.	Que el ícono de DashAI rediriga al menú principal y que los experimentos tengan colores distintivos.	9
Usuario 5	Buena experiencia.	Ninguna.	Fácil de seguir.	Si, pero el detalles de los experimento no es tan claro.	Equilibrado tirado a simple.	Si.	Capacidad de validar modelos con nuevos datos y la inclusión de gráficos para comparar los modelos de forma más visual, ya que considera que agregar y eliminar columnas puede resultar repetitivo.	Que el ícono de DashAI rediriga al menú principal.	7

F.1. RECOMENDACIONES DE LA CARGA DE DATOS Y EXPERIMENTOS

Carga de datos	Experimentos
Interfaz -Cuando suceda un error, indicar cómo solucionarlo. -Indicar el progreso de carga de un dataset. -"Humanizar" las palabras. De TabularClassificationTask a Tabular Classification. -Añadir un color distinto a cada tipo de tarea que tenga para poder diferenciarla. -Indicar cuántos datos van a haber en cada split (train, test y val). -Añadir un resumen estadístico del dataset. -Añadir un botón sobre un dataset para poder iniciar un experimento de manera rápida. Parámetros y carga de datos -Permitir el split de val en 0. -Permitir crear un tag a un dataset. -Permitir cargar datasets desde la web (Kaggle, HF, etc). -Añadir una alerta sobre valores atípicos o desequilibrados. Manipulación de datos -Permitir el uso de OneHotEncoder, normalizar y min-max scaler para escalar los datos. -Permitir centrar los datos entre -1 y 1. -Permitir filtrar filas del dataset. -Permitir al usuario la opción de ignorar la primera fila o no.	Interfaz -Añadir un botón de Finished que cierre la ventana. -Añadir un botón de Finished que dirija hacia los resultados del experimento. -Añadir un botón para pasar de experimentos a resultados. -Agrandar el botón de iniciar experimento. -Cambiar el ícono de update por otro más reconocible. -Indicar a qué columna se está haciendo la predicción. -Añadir una barra de progreso durante la ejecución (iteración actual y tiempo transcurrido). Ejecución modelo -Permitir entrenar otro modelo mientras se espera que finalice uno. -Al cancelar un experimento, entregar un modelo con la situación actual. -Añadir un botón para detener el modelo en ejecución. Otros -Permitir ejecutar un modelo varias veces (versiones). -Permitir crear un tag a un experimento. Modelos faltantes -Mostrar modelos más populares para el tipo de Task. -Añadir el modelo de redes neuronales. -Subir un modelo pre-entrenado en formato Keras. Manipulación datasets y métricas -Filtrar filas del dataset para un modelo en específico. -Seleccionar qué métricas serán visualizadas en "Results". -Añadir un botón para manejar las columnas de la tabla.

F.2. RECOMENDACIONES DE LOS RESULTADOS Y EXPERIENCIA GENERAL

Resultados	Experiencia general
Interfaz -Modificar el color del menu izquierdo que no se ve tan claramente, ya que se confunde con el fondo. -Modificar la explicación de los parámetros en la vista general y detallada. -Utilizar un ícono a la vez, no una combinación (detalles experimento). -En los resultados, indicar a qué corresponde el número "Status". -Los experimentos con error deben indicar qué sucedió y cómo solucionarlo. -Solo mostrar los parámetros que sí se utilizaron en ese modelo. -Diferenciar los parámetros por defecto y los modificados por el usuario. -En detalles de experimento - métricas, mostrar Test, Train y Val juntos Datos -Visualizar cuáles fueron los datos erróneos en la validación. -Indicar cuántas filas fueron correctamente clasificadas (no porcentaje). -Obtener la cantidad de elementos usados en cada split (número, no porcentaje). -Validar modelo con datos nuevos. Exportar y replicar modelo -Permitir guardar el modelo en Pickle, Json o Keras. -Permitir replicar el modelo (exportar). Gráficos y comparaciones -Gráficos para comparar distintos modelos. -Gráficos para ver el rendimiento del modelo. -Comparar versiones de un mismo modelo.	Interfaz -Modificar los botones de subir y bajar para que sean más grandes. -Redirigir a la página de Inicio tras pulsar el ícono de DashAI. Documentación -Indicar en qué afectan los parámetros si es que se cambian. -Mostrar los límites de los parámetros (desde un valor hasta otro). -Añadir un enlace a la documentación de sklearn. Practicar con DashAI -Añadir una módulo de ejemplos. -Añadir datasets de prueba. -Añadir experimentos de prueba (en base a un dataset de prueba). Qué es lo que hace el sitio -Añadir un tutorial que le muestre al usuario el flujo a seguir. -Incluir una explicación básica de qué es lo que hace la plataforma.

G.1. PRUEBA DE USABILIDAD DE LA VERSIÓN FINAL DEL PROTOTIPO

Introducción

<Nombre del usuario>, en primer lugar, quiero agradecerte por dedicar tu tiempo a participar en esta sesión de prueba de usabilidad para DashAI. Antes de iniciar, es importante brindarte un contexto sobre la aplicación que evaluarás. DashAI, desarrollada por el CENIA en conjunto con otras instituciones, se encuentra en una fase de pruebas con el propósito de realizar predicciones mediante AutoML a partir de *datasets* proporcionados por los usuarios. Actualmente, estamos trabajando en una interfaz que te presentaremos a continuación, buscando implementar mejoras basadas en las experiencias recogidas de pruebas de usabilidad como esta.

Durante la sesión, te invitamos a completar un conjunto de cuatro actividades y una encuesta de usabilidad, estimando un tiempo total de aproximadamente 30 minutos para el proceso. Te pedimos que realices estas tareas de manera independiente, como si no estuviese aquí para guiarte. Es crucial entender que el objetivo de esta sesión es evaluar el diseño de la aplicación, no al usuario. Por lo tanto, no hay respuestas correctas o incorrectas y no debes preocuparte por el resultado. Lo más importante es que, mientras avanzas en las tareas, compartas en voz alta tus pensamientos y experiencias.

Cualquier aspecto que te resulte confuso, frustrante o que suscite dudas, te animamos a comunicarlo. Tus observaciones son extremadamente valiosas y contribuirán significativamente a la mejora del diseño de DashAI.

[Se solicita permiso para grabar la sesión].

Entrevista inicial

¿En qué ciudad vives?

¿En qué proyectos o tareas has aplicado *machine learning*?

¿En qué metodología participaste en DashAI?

Contexto para realizar las tareas

Imagina que eres un científico de datos trabajando en un proyecto de investigación sobre el infarto de miocardio. Tienes un conjunto de datos que contiene la información de pacientes y te gustaría utilizar DashAI para analizar los datos y predecir con los resultados la probabilidad de que una persona tenga riesgo de sufrirlo. Quieres comparar diferentes modelos para decidir cuál es el más adecuado para tu estudio.

Tareas

Tarea 1: Tienes un archivo `Medicaldataset.csv` que contiene los datos médicos que necesitas cargar en DashAI, para lo cual buscas en la plataforma la opción de subir el

archivo y añadirlo a la lista de *datasets*. Posteriormente, analizas la información cargada para corroborar que los tipos de datos están correctos y son los necesarios. Usando la interfaz, fíjate si puedes subir este archivo y ver su resumen, recuerda pensar en voz alta y decirme cuando creas que la tarea ha sido finalizada.

Tarea 2: Ahora que tu *dataset* está cargado, quieres crear un experimento utilizando varios modelos para hacer predicciones. Navega por la interfaz para encontrar dónde y cómo puedes crear un nuevo experimento y posteriormente ejecutarlo.

Tarea 3: Tu experimento ha finalizado y estás ansioso por ver los resultados para entender el rendimiento de los modelos que has utilizado. Localiza y revisa los resultados de tu experimento en la plataforma.

Tarea 4: Para tomar la mejor decisión para tu proyecto de investigación, necesitas comparar el rendimiento de los modelos generados. Utilizando el experimento generado anteriormente, busca cómo puedes comparar los modelos en base a las distintas vistas que te otorga la interfaz de DashAI.

Entrevista final y agradecimientos

Primero que todo, agradecer de nuevo tu tiempo por asistir aquí para realizar la prueba.

Me gustaría saber tu opinión acerca de esta experiencia mediante la siguiente encuesta.

Gracias por venir.

G.2. HALLAZGOS

Usuario/Tarea	T1	T2	T3	T4
Usuario 1	El usuario asumió que el separador en el conjunto de datos era una coma, ya que no pudo revisar el <i>dataset</i> previamente.	Sintió confusión acerca de qué estaba tratando de predecir. Se confundió en la sección de <i>split</i> , ya que al principio ya había realizado esta división, pero ahora lo vuelven a preguntar y no sabe el por qué.	Asumió que el icono de visualización era para ver los datos. Pudo comparar los resultados utilizando la tabla y le gustó la visualización de gráficos. Sin embargo, encontró el gráfico de torta confuso, ya que no entendía qué representaba cada sección de la torta.	Sugiere separar las secciones de entrenamiento, validación y prueba de las métricas F1, precisión, exactitud y <i>recall</i> , ya que representan aspectos distintos del rendimiento del modelo. Encontró muy útil el poder elegir qué métrica elegir visualizar.
Usuario 2	Recuerda que era la misma vista de carga que probó en la prueba anterior y le gustó la sección de visualización de resúmenes. Sin embargo, está confundido acerca de si " <i>Example</i> " representa la cantidad de elementos o un ejemplo de esos elementos. Además, no comprende qué significa " <i>Column type Value</i> " ni qué indica esta columna. También pregunta qué sucede con los valores NaN en el caso del ejemplo proporcionado.	No recuerda las columnas que estaban presentes inicialmente y no está seguro si la selección predeterminada las muestra todas. A pesar de que identificó que habían nueve columnas en total, solo pudo visualizar cuatro en la carga.	Completó la tarea sin complicación.	El usuario interactuó con las opciones del gráfico y le agradaron. Sin embargo, al comparar varios modelos en el <i>Radar Plot</i> , no pudo apreciar claramente la comparación entre ellos. No entendió qué modelo correspondía a cada uno en el <i>Radar Plot</i> , ya que no seleccionó los nombres para los modelos. Aunque le gustaron los gráficos, prefirió la tabla de resultados, ya que la consideró más directa. No pudo encontrar el botón de métricas personalizadas.
Usuario 3	Completó la tarea sin complicación.	Apreció que haya una verificación en pantalla para asegurarse de que los datos que se van a trabajar coincidan con la <i>Task</i> . Sin embargo, encontró un problema con el <i>split</i> , ya que al presionar las flechas junto a los valores de <i>split</i> , estos aumentan en 1 en lugar de en 0.1 como deberían.	Le gustaron los gráficos, ya que permite visualizar rápidamente el <i>performance</i> .	Le gustó la parte interactiva de los gráficos.
Usuario 4	Desconoce si los datos están separados por comas y encuentra razonable la división entre los <i>splits</i> . Considera que sería útil definir la variable objetivo durante la carga de los datos. Asimismo, asume que todas las columnas son del tipo " <i>Value</i> ". No encuentra razonable que se le dé un ejemplo, ya que prefiere que se proporcionen estadísticas descriptivas más detalladas, como el promedio, la mediana, etc., que Pandas suele ofrecer.	Señala que cuando se cargaron los datos se definió el <i>split</i> , por lo que no está seguro si el nuevo <i>split</i> es del total de los datos o del experimento. Le gustaría tener una lista con todas las columnas, donde pudiera marcarlas con una marca de verificación o una cruz, para decidir cuáles incluir y cuáles no, y así determinar qué variables predecir y cuáles no. Además, prefiere ver el nombre de la columna en lugar de un número de entrada cuando realiza selecciones.	Se sorprendió con los gráficos, pero sugiere que sería útil conocer si los datos están desbalanceados o si hay una variable más importante para comprenderlos mejor. No encuentra sentido a los gráficos de torta, ya que no sabe qué son ni cómo interpretarlos. Aunque nunca había visualizado datos de esta manera y no le parece completamente claro, entiende que pueden llegar a ser útiles. En cuanto a las métricas personalizadas, las encuentra muy útiles y bien implementadas. Le gustaron los gráficos interactivos.	Indicó que solamente utilizaría el gráfico de barras, ya que visualmente es lo más cercano a lo que está acostumbrado a usar. Además, menciona que no recuerda qué significa el F1, lo que sugiere la necesidad de proporcionar una explicación o referencia sobre esta métrica. Sugiere que en la sección de conjuntos de datos, sería útil incluir una matriz de correlación para obtener una comprensión más completa de las relaciones entre las variables.
Usuario 5	Le gustaría tener más libertad para explorar los datos, ya que usualmente dedica la mayor parte de su tiempo a esta tarea.	Notó un error donde al modificar el valor de entrenamiento a 0.7 y el de validación a 0.2, y el de prueba a 0.1, las casillas se vuelven de color rojo, siendo que no debería ser así al sumar todas 1.	El gráfico de barras fue el que más le gustó, mientras que el de radar lo encontró atractivo y funcional. Encuentra que algunos gráficos son redundantes debido a los parámetros utilizados.	Considera que alguien que desarrolla el modelo encontraría útil la tabla, pero para alguien que no está familiarizado con ello, preferiría los gráficos. Le gustaría que los resultados tengan su módulo propio, ya que lo considera una sección importante a considerar.