



**Apoyando la evolución de FTVault: Resolución de anomalías y mejoras en la interfaz
de usuario**

Johan Alejandro Cristancho Medina

Informe presentado como requisito para optar al título de Ingeniero de Sistemas

Asesores:

José Ignacio López Vélez & Cesar Alejandro Arango Palacio

Modalidad:

Práctica empresarial

Universidad de Antioquia
Facultad de Ingeniería, Ingeniería de Sistemas
Medellín – Colombia
2023

Cita	Cristancho Medina [1]
Referencia Estilo IEEE (2020)	[1] J. A. Cristancho Medina, “Apoyando la evolución de FTVault: Resolución de anomalías y mejoras en la interfaz de usuario”, Trabajo de grado profesional, Ingeniería de Sistemas, Universidad de Antioquia, Medellín, Antioquia, Colombia, 2023.



Centro de Documentación Ingeniería (CENDOI)

Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - www.udea.edu.co

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

Dedicatoria

Con gratitud en mi corazón, dedico este logro en mi alma mater a Dios, mi familia y a mí, como un sincero recordatorio de este valioso proceso. Ha sido alcanzado con consistencia, disciplina y excelencia, y me ha enseñado que, con fe, apoyo y esfuerzo, puedo lograr cualquier meta que me proponga. Que esta dedicatoria perdure como un testimonio de mi crecimiento y determinación en el camino hacia el éxito.

Agradecimientos

A mi familia, por brindarme su incondicional apoyo. A mi alma mater, la Universidad de Antioquia, por nutrir mi proceso de aprendizaje y desarrollo. A Rockwell Automation, por la invaluable oportunidad de crecimiento profesional. Y a mis apreciados asesores, José Ignacio López Vélez y Cesar Alejandro Arango Palacio, por su invaluable guía en mi trayectoria.

Tabla de contenido

Tabla de contenido	4
Lista de tablas.....	6
Lista de figuras	6
Resumen	7
Abstract	8
Introducción	9
Situación problemática.....	11
Objetivos.....	12
Objetivo General	12
Objetivos Específicos	12
Metodología.....	13
Cronograma de actividades	14
Alcance.....	15
Marco teórico.....	16
Interfaz de usuario:.....	16
Experiencia de usuario:.....	17
Desarrollo frontend:.....	17
Metodologías ágiles:	17
Framework:	19
Connected enterprise:	19
Diseño Atómico:	20
Código limpio:	21
Usabilidad:	21
• Aprendizaje.....	21
• Eficiencia	21
• Memorabilidad	22
• Errores.....	22
• Satisfacción	22
Figma.....	22

Calidad de software.....	23
Requisitos funcionales	24
Requisitos no funcionales.....	24
Spike.....	24
Pruebas unitarias	25
Resultados y análisis	26
Conclusiones y trabajo futuro.....	38
Conclusiones.....	38
Trabajo futuro.....	39
Referencias	40

Lista de tablas

Tabla 1 Cronograma de sprints.....	14
Tabla 2 Tickets realizados durante la práctica.....	27
Tabla 3 Conteo por categoría.....	30

Lista de figuras

Imagen 1 Ejemplo de interfaz de usuario - wireframe	16
Imagen 2 Ejemplo ticket de Jira – herramienta de gestión.....	18
Imagen 3 Metodología diseño atómico	20
Imagen 4 Prototipo Ejemplo de Figma.....	23
Imagen 5 Gráfico de cantidad de tickets por categoría	31
Imagen 6 Gráfico de puntos de historia por categoría.....	32
Imagen 7 Anomalía de UI-UX (4233).....	33
Imagen 8 Anomalía de UI-UX (3895) parte 1.....	34
Imagen 9 Anomalía de UI-UX (3895) parte 2.....	34
Imagen 10 Optimización - estado inicial (4157)	35
Imagen 11 Optimización - estado final (4157).....	35
Imagen 12 Tarea de UI-UX (3863)	36
Imagen 13 Resultados tests unitarios al inicio de la práctica	36
Imagen 14 Resultados tests unitarios al final de la práctica	37

Resumen

En este proyecto colaborativo con el equipo de desarrollo del producto FTVault de Rockwell Automation se lograron exitosamente los objetivos de mejorar la experiencia e interfaz de usuario a través de un proceso que incluyó el uso de metodologías ágiles, validación de código, pruebas unitarias y una implementación adecuada. Durante este proceso, se abordaron tareas relevantes, especialmente en el área de UI-UX, fortaleciendo la calidad del producto y satisfaciendo las necesidades de los usuarios.

La incorporación de nuevas funcionalidades, resolución de anomalías y de más mejoras no solo aportó valor al producto, sino también enriqueció la experiencia general. Se dio especial importancia a los requisitos no funcionales, considerados como métricas de calidad del producto, para garantizar la fluidez y amabilidad en la interfaz con el usuario. Esta dedicación a la experiencia del usuario, junto con la optimización del producto FTVault, ha sido fundamental en su evolución y posicionamiento en el mercado, asegurando la satisfacción de los clientes y el éxito general del proyecto.

Palabras claves: Desarrollo frontend, Experiencia de Usuario (UX), Interfaz de Usuario (UI), metodologías ágiles y Connected Enterprise.

Abstract

In this collaborative project with the product development team of Rockwell Automation's FTVault, the objectives of improving the user experience and interface were successfully achieved through a process that included the use of agile methodologies, code validation, unit testing, and appropriate implementation. Relevant tasks, particularly in the UI-UX area, were addressed during this process, strengthening the product's quality and meeting user needs.

The addition of new features, resolution of anomalies, and further improvements not only added value to the product but also enhanced the overall experience. Special attention was given to non-functional requirements, considered as quality metrics for the product, to ensure smoothness and user-friendliness in the interface. This dedication to the user experience, coupled with the optimization of the FTVault product, has been instrumental in its evolution and positioning in the market, ensuring customer satisfaction and overall project success.

Keywords: Frontend Development, User Experience (UX), User Interface (UI), Agile Methodologies, Connected Enterprise.

Introducción

En el mundo empresarial actual, el éxito de los productos de software se basa en la adopción de nuevas metodologías y la consideración de diversos aspectos. Entre ellos, es crucial cumplir tanto con los requisitos funcionales como con los no funcionales, siendo estos últimos los más cercanos al cliente y utilizados como métricas para medir la calidad del producto. En este contexto, el objetivo de esta práctica académica es agregar nuevas funcionalidades y resolver anomalías del sistema siguiendo la metodología actual de la empresa Rockwell Automation.

Rockwell Automation (RA), es la empresa detrás del desarrollo del producto de software FTVault, y considerada como líder global en soluciones de automatización industrial. Fundada en 1903, RA ofrece una amplia gama de productos y servicios que ayudan a las empresas a optimizar y mejorar sus procesos de producción. Su experiencia en la industria y su enfoque en la innovación tecnológica han establecido a RA como un referente en la automatización y control industrial, brindando soluciones confiables y eficientes a sus clientes en todo el mundo.

Ahondando un poco más en el producto sobre el cual se realizarán los aportes, FTVault es una innovadora solución de software que actúa como un cofre virtual, permitiendo a las empresas almacenar de forma segura información vital para sus procesos de producción, como configuraciones de equipos, datos de producción, informes de mantenimiento y otros documentos importantes. Gracias a su robusto sistema de seguridad, el software garantiza la integridad y confidencialidad de los datos almacenados, protegiéndolos contra accesos no autorizados y posibles pérdidas. Su aplicación versátil se adapta a diferentes sectores de la industria, brindando un enfoque personalizado para satisfacer las necesidades específicas de cada cliente.

Bajo esta premisa de confiabilidad y eficiencia, esta práctica académica se enfoca en el desarrollo y mejora del producto FTVault. Durante este proceso, se incorporarán nuevas funcionalidades para enriquecer la experiencia de los usuarios y se abordarán las anomalías identificadas en el sistema. Todo ello se llevará a cabo siguiendo la metodología actual de la

empresa, garantizando así que el software cumpla con los más altos estándares de calidad y se alinee perfectamente con las necesidades y expectativas de los usuarios. El objetivo es lograr un producto de excelencia, que satisfaga tanto los requisitos funcionales como los no funcionales, y que brinde un valor significativo tanto para los usuarios como para la organización en su conjunto.

Situación problemática

La situación problemática que se enfrenta en el contexto actual es la necesidad de cumplir con los requisitos tanto funcionales como no funcionales en los productos de software. Con la amplia oferta de soluciones disponibles en el mercado, las empresas deben adoptar nuevas metodologías y considerar diversos aspectos para asegurar la calidad de sus productos. Es especialmente importante prestar atención a los requisitos no funcionales, que están directamente relacionados con la experiencia del cliente y la calidad del producto, por lo que resultan relevantes en términos de su satisfacción.

En el caso específico del producto FTVault, se ha identificado la presencia de anomalías tanto en su funcionamiento como en la experiencia e interfaz de usuario. Estas anomalías representan obstáculos y limitaciones que afectan la eficiencia y la satisfacción de los usuarios. Por ejemplo, podría haber posibles mejoras de rendimiento, fluidez en la navegación o en el acceso a la información almacenada por medio de la interfaz de usuario.

El abordar estos aspectos permite incrementar la productividad y la eficacia de los equipos que utilizan el FTVault. La falta de funcionalidades adicionales también limita las posibilidades de optimización y personalización de la solución según las necesidades específicas de los usuarios. Por tanto, es crucial abordar estas anomalías y agregar nuevas funcionalidades para mejorar la experiencia del usuario, aumentar la eficiencia en el acceso y gestión de los diseños, y ofrecer herramientas de análisis más completas. Todo ello contribuirá a elevar la calidad y la competitividad del producto en el mercado empresarial.

Objetivos

Objetivo General

Colaborar al equipo de desarrollo *frontend* en el producto FTVault por medio de la elaboración de tareas requeridas siguiendo la metodología del proyecto, con el principal objetivo de mejorar las funcionalidades tanto actuales como nuevas y la experiencia e interfaz del usuario.

Objetivos Específicos

- Identificar y priorizar oportunidades de mejora en el desarrollo *frontend* de FTVault para proporcionar una mejor experiencia de usuario a los clientes de la empresa Rockwell Automation.
- Desarrollar e implementar nuevas funcionalidades o mejoras en el producto FTVault basándose en las oportunidades de mejora priorizadas y que aporten más valor a los usuarios.
- Contribuir en la documentación y pruebas necesarias para garantizar la calidad del producto FTVault, siguiendo la metodología de la empresa y cumpliendo los requisitos para cada tarea a realizar.

Metodología

Para garantizar la eficacia y eficiencia en el proceso de desarrollo de software, es fundamental contar con metodologías y herramientas adecuadas. En el equipo de desarrollo *frontend* para este producto, se ha adoptado Scrum como metodología de gestión de proyectos, Jira como herramienta de gestión, Angular como *framework* de desarrollo para aplicaciones web, y Figma como herramienta de diseño y prototipado de interfaces. Esta combinación de herramientas y metodología permite trabajar de manera colaborativa, planificar y llevar a cabo entregas de software más rápidas y eficientes, y mantener la calidad del desarrollo de aplicaciones.

Las actividades necesarias seguidas por la empresa para el cumplimiento de objetivos son:

1. Planificación del *Sprint*.
2. Daily Scrum.
3. Desarrollo de tareas.
4. Reuniones de Revisión y Retrospectiva del *Sprint* (véase en el siguiente ítem las fechas).
5. Documentación y pruebas.

Cronograma de actividades

Tabla 1 Cronograma de sprints.

Actividad	Fecha de inicio	Fecha fin
FY 2023 - Q2 Sprint 45th	30/01/2023	16/02/2023
FY 2023 - Q2 Sprint 46th	20/02/2023	02/03/2023
FY 2023 - Q2 Sprint 47th	06/03/2023	23/03/2023
FY 2023 - Q3 Sprint 48th	03/04/2023	20/04/2023
FY 2023 - Q3 Sprint 49th	24/04/2023	11/05/2023
FY 2023 - Q3 Sprint 50th	15/05/2023	01/06/2023
FY 2023 - Q3 Sprint 51th	05/06/2023	22/06/2023
FY 2023 - Q2 Sprint 45th	30/01/2023	16/02/2023

Nota: Cada *sprint* consta de las diferentes actividades mencionadas en el ápice anterior, fecha de inicio se toma como la planificación y la de cierre es la demostración de lo conseguido durante el *sprint*. Cabe aclarar que se realiza una reunión de retrospectiva, usualmente al día siguiente de finalizar el *sprint*.

Alcance

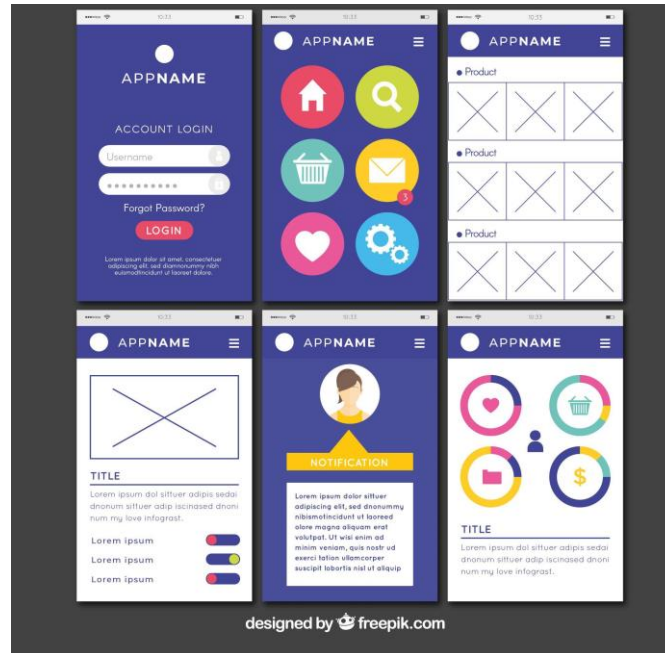
Los resultados esperados del trabajo como practicante de desarrollo *frontend* en el proyecto FTVault son:

- Mejora en la experiencia del usuario para simplificar la interacción con las aplicaciones y soluciones de Rockwell Automation, resolviendo las anomalías detectadas que se consideren relevantes de abordar en el proyecto (tanto de funcionamiento, estilo o flujo del sistema).
- Implementación de nuevas funcionalidades y mejoras, que sean consideradas por el equipo relevantes para mantener la solución actualizada y adaptada a las necesidades de los clientes de Rockwell Automation.
- Contribución en la documentación y pruebas unitarias garantizando que toda tarea realizada cumpla con la meta del 80% de cobertura de código, asegurando la calidad de las nuevas funcionalidades del producto FTVault.

Marco teórico

Interfaz de usuario:

Imagen 1 Ejemplo de interfaz de usuario - wireframe



Nota: Tomada de freepik.com

La interfaz de usuario (UI, por sus siglas en inglés) es la parte de un programa o aplicación que nos permite interactuar con él de manera visual. Es como la cara del software, lo que vemos en la pantalla y con lo que podemos hacer clic, escribir o mover. La UI se preocupa de cómo se ve y se siente el producto, más que de cómo funciona. Su objetivo es hacer que la interacción con el programa sea lo más sencilla y eficiente posible para que los usuarios puedan lograr sus objetivos de manera fácil. Para lograrlo, se fusionan las necesidades del usuario con el diseño visual, y el resultado es una representación gráfica del producto llamada *wireframe* (Imagen 1) de alta resolución. [1]

Experiencia de usuario:

La experiencia de usuario (UX, *por sus siglas en inglés*) se refiere a cómo nos sentimos al interactuar con un producto o servicio. Es la vivencia que tenemos al utilizar el programa, que incluye aspectos como la facilidad de uso, la accesibilidad y el placer que experimentamos durante la interacción. El Diseño de Experiencia de Usuario busca mejorar la satisfacción del usuario en cada paso del proceso, considerando factores emocionales y funcionales. Se basa en una investigación profunda de usuarios para crear *wireframes* de baja resolución que integren las necesidades del usuario, la visión empresarial y la factibilidad tecnológica, logrando así que el diseño se conecte de manera significativa con las expectativas y deseos de los usuarios. El objetivo es proporcionar una experiencia gratificante y valiosa que haga que el usuario disfrute y se sienta cómodo utilizando el producto o servicio. [1]

Desarrollo frontend:

El desarrollo *frontend* se enfoca en crear la parte visual e interactiva de un sitio web o aplicación, con el objetivo de ofrecer una experiencia de usuario efectiva. Los desarrolladores *frontend* utilizan tecnologías como HTML, CSS y JavaScript para diseñar la estructura y formato de las páginas web, añadir interactividad a la interfaz y mejorar la experiencia del usuario.

Cabe destacar que el desarrollo *frontend* se enfoca en mejorar la interacción de los usuarios con una aplicación o sitio web mediante la creación de interfaces de usuario intuitivas y atractivas. Los desarrolladores *frontend* utilizan diferentes tecnologías para diseñar y dar formato a la interfaz de usuario, y aseguran que sea compatible con diferentes dispositivos y navegadores para garantizar una experiencia de usuario óptima. [2]

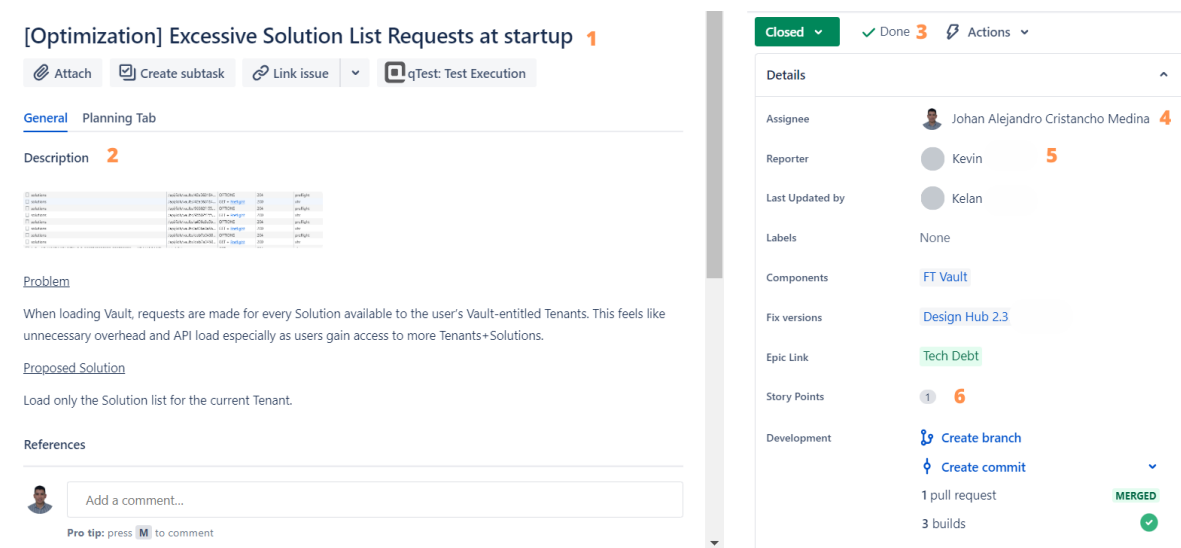
Metodologías ágiles:

La metodología ágil es un enfoque colaborativo y adaptable que busca lograr una entrega eficiente de productos. Para hacerlo, se utilizan puntos de historia que estiman la complejidad de las tareas y se asignan a tickets (imagen 2) que representan funcionalidades

específicas a abordar en cada *sprint*, que es un período de trabajo de 2 a 3 semanas. Estos tickets se gestionan en un panel Kanban o Scrum, siguiendo un flujo de trabajo que incluye los estados de: "por hacer", "en progreso", "en revisión" y "cerrado". Durante el *sprint*, el equipo realiza reuniones diarias para hacer seguimiento del avance de las tareas, resolver dudas o bloqueos, y al final del *sprint*, se realizan revisiones y retrospectivas para mejorar el proceso. Esta metodología permite a los equipos adaptarse rápidamente a los cambios y recibir retroalimentación continua de los clientes, lo que facilita el desarrollo de productos que se ajusten a las necesidades y expectativas del mercado.

En este contexto, el rol del *Product Owner (PO)* es esencial ya que representa los intereses del cliente y desempeña un papel crucial en la definición y priorización de los requisitos del producto. Trabaja estrechamente con el equipo de desarrollo, toma decisiones sobre el alcance y la dirección del proyecto, y asegura que se cumplan las expectativas del cliente. El *Product Owner* tiene la responsabilidad de maximizar el valor entregado al cliente y mantener un enfoque centrado en el usuario. Su colaboración efectiva con el equipo de desarrollo y la toma de decisiones acertadas contribuyen al éxito del proyecto y a la entrega de un producto de calidad. [3]

Imagen 2 Ejemplo ticket de Jira – herramienta de gestión



Nota: se enumera de 1 al 6 las secciones importantes

Como se puede ver en la imagen 2, se han agregado una enumeración de 1 a 6 para identificar las secciones relevantes dentro de un *ticket*. La sección uno representa el título de la tarea, la dos corresponde a la descripción, donde se recomienda añadir pantallazos o en caso de *tickets* de UI-UX el diseño a implementar. Continuando con las secciones, en la número tres se encuentra el estado de la tarea, en este caso ‘*closed*’ ya que está tarea ha sido cerrada. En la cuatro quién la está o la ha trabajado, la cinco quien la reportó, en otras palabras, quien considera que esto es relevante de abordar. Para finalizar, la sección seis donde se establecen los puntos de historia, medida que sigue la serie Fibonacci y busca estimar la complejidad de la tarea.

Framework:

Un *framework* web es una herramienta de desarrollo de software que proporciona una estructura y conjunto de herramientas para simplificar el proceso de creación de aplicaciones web. Está diseñado para facilitar la tarea de escribir y mantener aplicaciones web complejas al proporcionar un conjunto de patrones de diseño, reglas y estándares comunes que los desarrolladores pueden seguir. Los *frameworks* web también incluyen una serie de bibliotecas y componentes reutilizables que los desarrolladores pueden utilizar para construir aplicaciones web rápidamente y con mayor eficiencia. [4]

Connected enterprise:

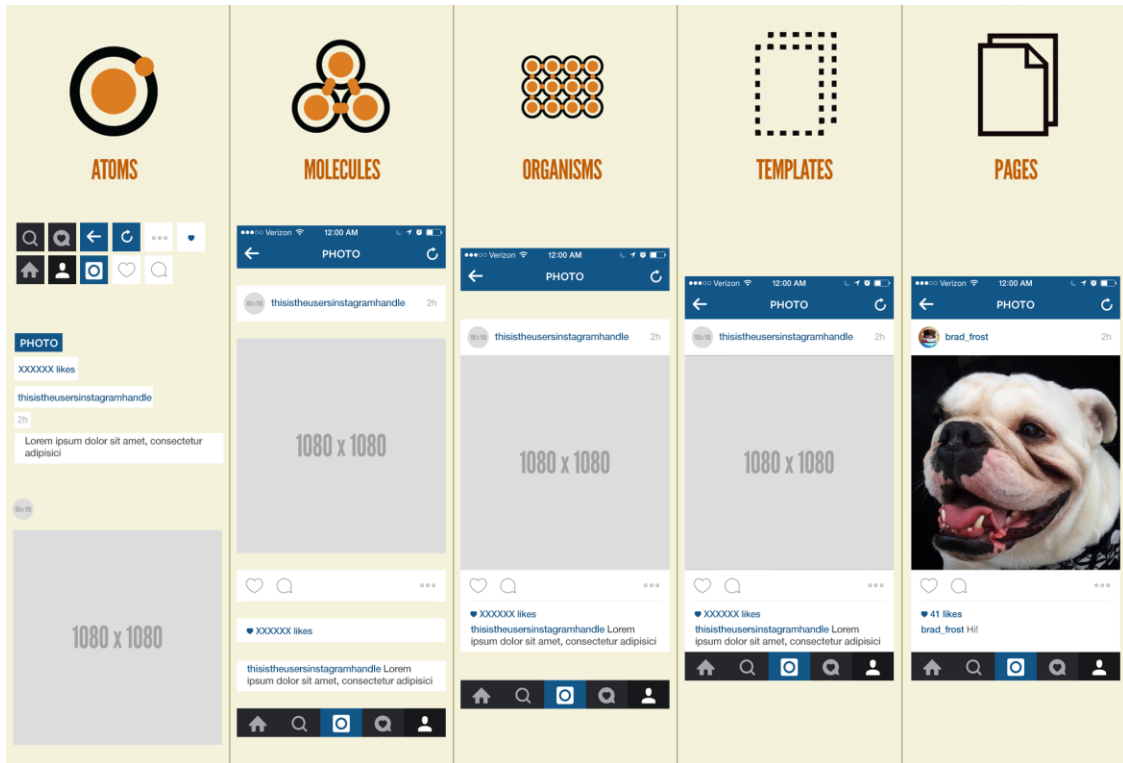
De acuerdo con la definición de este concepto por parte de la empresa Rockwell Automation en su sitio web.

“La fabricación inteligente es la puerta hacia la transformación digital. Los dispositivos inteligentes conectados abren ventanas nuevas hacia la visibilidad de los procesos. Los datos y el análisis permiten tomar decisiones mejores y más rápido. La conectividad transparente incentiva nueva colaboración.

The Connected Enterprise® hace todo esto posible. Reúne redes empresariales y a nivel de la planta y conecta en forma segura a personas, procesos y tecnologías.” [5]

Diseño Atómico:

Imagen 3 Metodología diseño atómico



Nota: Imagen tomada del blog de Brad Frost donde explica el diseño atómico [6].

El diseño atómico, es una metodología que descompone las interfaces de usuario en componentes más pequeños y reutilizables llamados átomos. Estos átomos se combinan para formar moléculas y luego organismos, creando componentes más complejos. Además, se utilizan plantillas y páginas para proporcionar una estructura y contenido específico.

La filosofía del diseño atómico es fomentar la modularidad, la reutilización y la consistencia en el diseño. Al descomponer las interfaces en elementos más pequeños, se facilita el desarrollo y mantenimiento de sistemas de diseño coherentes, lo que conduce a una mayor eficiencia, coherencia visual y una mejor experiencia de usuario en general. Como se

evidencia en la imagen 3, el diseño atómico permite a los equipos construir sistemas escalables y flexibles, promoviendo la coherencia en todas las aplicaciones y facilitando la evolución del diseño a medida que se agregan nuevas funcionalidades o se actualizan los diseños. Cabe resaltar que se hace una demostración del uso de estos conocimientos, haciendo la abstracción de componentes para la red social Instagram. [6]

Código limpio:

Código limpio o *Clean Code* se refiere a la práctica de escribir código de software que sea legible, comprensible y fácil de mantener. Se trata de seguir principios y pautas que promueven la claridad, la simplicidad y la eficiencia en la escritura del código. El código limpio se caracteriza por ser fácilmente entendible por otros desarrolladores, lo que facilita la colaboración y la detección de errores. Además, el código limpio busca minimizar la complejidad innecesaria, utilizar nombres de variables y funciones descriptivas, evitar la duplicación de código y seguir convenciones de estilo consistentes. La adopción de prácticas de *Clean Code* no solo mejora la calidad del software, sino que también facilita su mantenimiento a largo plazo y permite realizar modificaciones y mejoras de manera eficiente. [7]

Usabilidad:

La usabilidad es un atributo de calidad que evalúa qué tan fácil es utilizar las interfaces de usuario. Este término también se refiere a los métodos para mejorar la facilidad de uso durante el proceso de diseño. La usabilidad puede ser definida mediante la respuesta a las preguntas que fundamentan los 5 componentes de calidad:

- **Aprendizaje**

¿Qué tan fácil es para los usuarios llevar a cabo tareas básicas la primera vez que se encuentran con el diseño?

- **Eficiencia**

Una vez que los usuarios han aprendido el diseño, ¿con qué rapidez pueden realizar tareas?

- **Memorabilidad**

Cuando los usuarios regresan al diseño después de un período de no utilizarlo, ¿qué tan fácilmente pueden recuperar su habilidad?

- **Errores**

¿Cuántos errores cometen los usuarios, qué tan graves son estos errores y qué tan fácilmente pueden recuperarse de ellos?

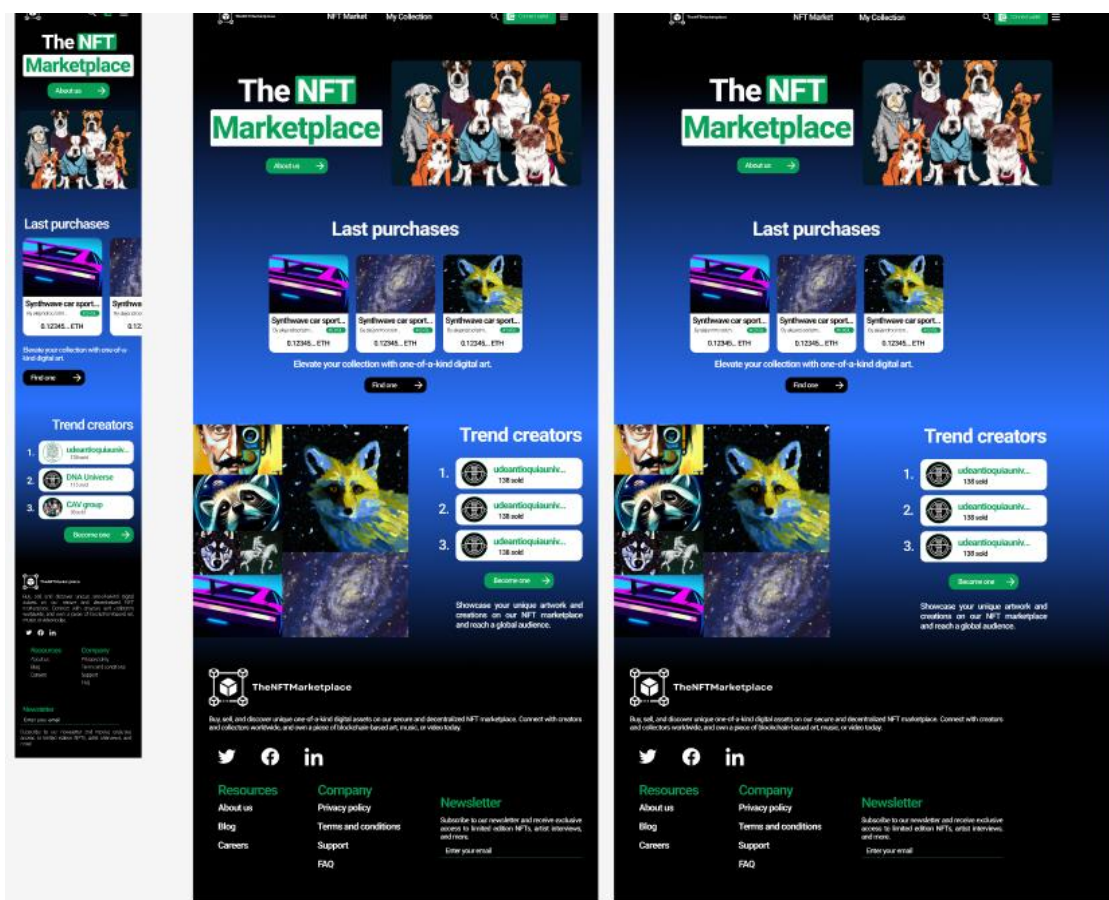
- **Satisfacción**

¿Qué tan agradable es utilizar el diseño? [8].

Figma

Figma es una herramienta de diseño y prototipado en la nube ampliamente utilizada en la industria del software. Proporciona una interfaz fácil de usar y potentes funciones de diseño que permiten a los usuarios crear diseños interactivos y prototipos de productos de software. Una de las principales ventajas de Figma es su capacidad de colaboración en tiempo real, lo que facilita el trabajo en equipo y la comunicación entre diseñadores, desarrolladores y otros miembros del equipo. Con Figma, los equipos pueden crear, compartir y revisar diseños como el de la imagen 4 de manera eficiente, lo que resulta en una mejora en la productividad y la calidad del producto final. [9]

Imagen 4 Prototipo Ejemplo de Figma



Nota: Elaboración propia para un proyecto universitario

Calidad de software

La calidad del software se refiere a las características que determinan su utilidad y excelencia, como eficiencia, flexibilidad, confiabilidad y seguridad. Para obtener software de calidad, es necesario utilizar metodologías estándares en el análisis, diseño, programación y prueba, siguiendo una política sustentada en principios tecnológicos, administrativos y ergonómicos. Además, se requiere un control continuo de la calidad mediante la definición de parámetros de medición y la implementación de procesos de evaluación. Esto asegura que

el software cumpla con los estándares, sea confiable y se mantenga flexible durante su ciclo de vida. [10]

Requisitos funcionales

Los requisitos funcionales describen las funciones y las acciones específicas que debe llevar a cabo el software. Estos requisitos se centran en qué tareas o acciones debe ser capaz de realizar el sistema. Representan el comportamiento y las funcionalidades esperadas del software. Los requisitos funcionales suelen estar asociados con las interacciones entre el sistema y los usuarios, así como con otros sistemas o componentes. [11]

Requisitos no funcionales

Los requisitos no funcionales se refieren a atributos y características del software que no están directamente relacionados con las funciones específicas. Estos requisitos se centran en aspectos como el rendimiento, la seguridad, la usabilidad, la fiabilidad, la escalabilidad, entre otros. Los requisitos no funcionales describen las restricciones y las cualidades deseadas del sistema, que son igualmente importantes para su correcto funcionamiento y calidad. [11]

Del mismo modo, los requisitos no funcionales son esenciales para medir la calidad del software, ya que aseguran que el software no solo cumpla con sus funciones principales, sino que también proporcione una experiencia positiva y satisfactoria para los usuarios, sea seguro, fiable, eficiente y adaptable a diferentes situaciones. Una aplicación que cumpla con estos requisitos será percibida como de alta calidad y será preferida por los usuarios en comparación con aquellas que no los satisfagan completamente. [11]

Spike

En el desarrollo de software, un *spike* es una investigación técnica breve y puntual para abordar problemas complejos o encontrar soluciones. Comúnmente se realizan *spikes* cuando existe incertidumbre en el enfoque de un problema o una funcionalidad. Se asigna un tiempo limitado, como un día o una semana, para investigar y experimentar diferentes enfoques. Al finalizar el spike, se espera obtener información suficiente para tomar decisiones informadas.

Los *spikes* también son útiles para explorar tecnologías emergentes, evaluar herramientas o *frameworks*, y resolver problemas técnicos complejos. En resumen, los *spikes* son investigaciones breves que abordan problemas técnicos, reducen la incertidumbre y proporcionan información relevante para la toma de decisiones en el desarrollo de software. [12]

Pruebas unitarias

Las pruebas unitarias son una técnica fundamental en el desarrollo de software que se utiliza para verificar el comportamiento correcto de unidades individuales de código, como funciones, métodos o clases. Estas pruebas se enfocan en aislar y probar cada unidad de forma independiente, ayudando a detectar errores tempranamente, mejorar la calidad del software y proporcionar confianza en el funcionamiento correcto de las unidades de código probadas. Al ejecutar pruebas unitarias de manera automatizada, los desarrolladores pueden ahorrar tiempo y recursos, facilitar la comprensión del código y garantizar la estabilidad del software en el proceso de desarrollo. [13]

Resultados y análisis

En esta sección, se presenta el análisis de los resultados obtenidos durante la práctica de desarrollo en el producto FTVault. Se utilizaron metodologías ágiles, como Scrum, para planificar y abordar los *tickets* en cada *sprint*, priorizando las oportunidades de mejora. A través de un enfoque colaborativo, se trabajó en la solución de las anomalías identificadas y en la implementación de nuevas funcionalidades que agregarán valor a los usuarios.

Durante el proceso, se llevaron a cabo reuniones diarias de seguimiento y revisiones periódicas para evaluar el progreso y realizar ajustes en el desarrollo. Al finalizar cada *sprint*, se realizó una revisión exhaustiva del trabajo completado, incluyendo pruebas internas para garantizar la correcta resolución de las anomalías y la funcionalidad de las nuevas características implementadas. Cabe destacar que cada *ticket* que involucra UI-UX pasa por la revisión del *product owner*, siendo este rol fundamental para la validación de los cambios realizados.

En el contexto de los objetivos específicos de la práctica, se realizaron diversas tareas para identificar y priorizar oportunidades de mejora en el desarrollo *frontend* de FTVault y proporcionar una mejor experiencia de usuario a los clientes de la empresa RA. Algunas de estas tareas fueron identificadas internamente, mientras que otras fueron descubiertas y priorizadas por el equipo de desarrollo, *testers* o el *Product Owner* del proyecto.

Entre las oportunidades de mejora identificadas, se encontraron anomalías en el área de botones de opción en el historial de archivos del proyecto, donde se observó una falta de consistencia en el tamaño de los botones. También se detectó un cambio que afectaba al botón de guardado al cambiar roles en el sistema. Estas anomalías, identificadas por el practicante como parte de las tareas abordadas durante la práctica, fueron resueltas para mejorar la experiencia de usuario (marcadas en la Tabla 2. columna de clasificación con [I]).

Es importante destacar que, además de las tareas internamente identificadas, también se trabajó en otras oportunidades de mejora que fueron priorizadas por el equipo de desarrollo, *testers* o el *Product Owner*. Estas oportunidades pueden haber surgido de la

evaluación del impacto generado por el proyecto o de las necesidades específicas de los usuarios. El enfoque conjunto en la identificación y priorización de estas oportunidades contribuyó a asegurar la calidad y la mejora continua del producto FTVault.

Posterior a esta identificación de tareas y sus prioridades, se continúa con el segundo objetivo específico, la implementación. Durante los *sprints* que conformaron la etapa de implementación de esta práctica académica se abordaron los *tickets* de la Tabla 2.

Tabla 2 Tickets realizados durante la práctica

Id	Título	Puntos de historia	Clasificación
4133	[UI] Agregar información emergente para acciones desactivadas	2	UI-UX
4208	[UI] Ordenar alfabéticamente la lista de proyectos en el panel de propiedades de la solución	2	UI-UX
3863	[UI] Implementar advertencias en la herramienta de comparación	5	UI-UX
4233	[UI] Ampliar área cliqueable para selección de solución	1	Anomalía de UI-UX
3895	El filtro por estado desde proyectos no funciona en español	2	Anomalía de UI-UX
2153	[Logging] UI: Convertir uso de console.log al nivel apropiado del registro	2	Desarrollo

4157	[Optimización] Solicitudes excesivas de la lista de soluciones al iniciar	1	Optimización
4391	[UI] Cambiar de Tenants en la página de análisis debería dirigirse al almacén de esa organización	2	Anomalía de UI-
4316	[DevTools] Indicador de estado LED	3	Desarrollo
4398	Diferentes estados para proyectos que no identifican la versión de Logix: historial y barra lateral de propiedades	1	Anomalía de UI-UX [I]
4477	El archivo con nombre en el panel del depósito de FT muestra unused.ZIP en lugar del nombre correcto del archivo del proyecto.ZIP	2	Anomalía de UI-UX
4715	El área de botones de opción en el historial de archivos del proyecto es más grande que el botón	1	Anomalía de UI-UX [I]
40	Fuga de memoria en notificaciones masivas	1	Optimización
46	[Dev] Mejorar la carga de un nuevo archivo al historial del proyecto	5	UI - UX [I]

102	Localización: agregar opciones de idiomas adicionales	2	UI - UX
113	[UI] El combo de comparación de rutinas debería estar desactivado con una sola entrada en el historial del proyecto	1	Anomalía de UI-UX
99	[UI] El icono de actualización de FTDS interrumpe la alineación dentro de los mosaicos del depósito	1	Anomalía de UI-UX
76	[UI] La propiedad HistoryUpdates.recordsUpdated no se utiliza en HistoryFilesComponent	1	Desarrollo
353	Falta el botón de guardar al cambiar roles	1	Anomalía de UI-UX [I]
392	[UI] Eliminar la verificación premium de la pestaña de gestión de usuarios	2	UI - UX
66	[Spike] Activar las "Acciones" del depósito dentro de la interfaz de usuario de Gitea	Spike Todo el sprint	Spike

Al realizar la sumatoria de la Tabla, se encuentra que fueron asignados en total 38 puntos de historia, de los cuales se logró cumplir el 100% satisfactoriamente. Cabe resaltar que la cuarta columna clasifica los *tickets* según su enfoque, ya que se encontró que podían agruparse en las siguientes etiquetas: Spike, Desarrollo, Optimización, UI - UX y Anomalía de UI - UX.

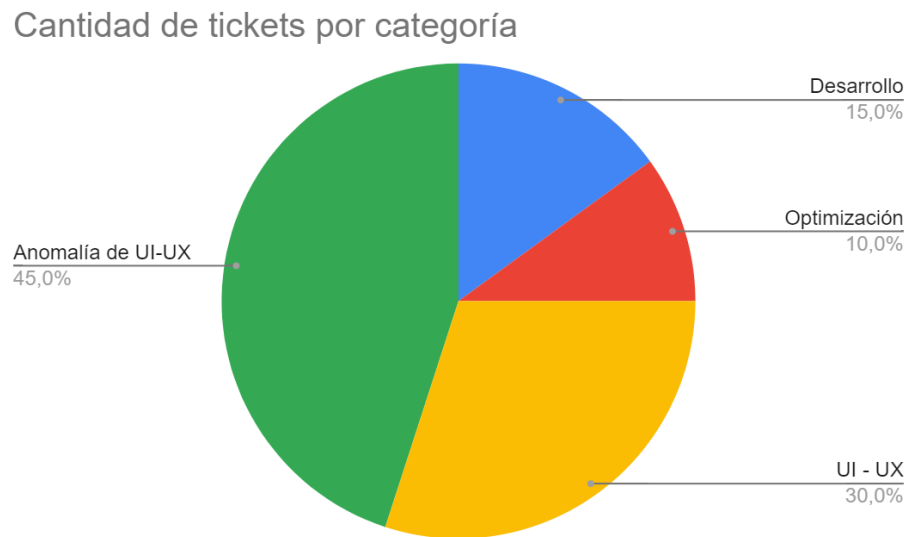
- **Spike:** Durante el único spike que fue realizado, se evaluó la posibilidad de utilizar Gitea como proveedor de servicios git para nuevas funcionalidades que se encuentran en fase de pruebas para el producto de FTVault.
- **Desarrollo:** Para este apartado, se hace referencia a los *tickets* relacionados a la mejora en el código o en el proceso de desarrollo para futuras implementaciones. Bajo esta etiqueta encontramos inserción de servicios para mostrar mensajes en la consola, leds sobre conexiones relevantes para los desarrolladores del producto y limpieza de código.
- **Optimización:** Dentro de los *tickets* que se encuentran relacionados al proceso de optimización se puede ver la mejora en el uso de los recursos y la reducción de peticiones innecesarias que pueden afectar el rendimiento.
- **UI - UX:** Del mismo modo en el que se han venido explicando los términos de UI y UX se categorizan dentro de esta etiqueta los *tickets* relacionados a la implementación de nuevas funcionalidades dentro de la interfaz de usuario y mejoras en la experiencia de usuario.
- **Anomalía de UI - UX:** Con alta relación a la anterior categoría se encuentran las irregularidades dentro de las funcionalidades ya implementadas o en la experiencia de usuario.

Tabla 3 Conteo por categoría

Categoría	Cantidad de tickets	Puntos de historia
Spike	1	-
Desarrollo	3	6
Optimización	2	2
UI - UX	6	18
Anomalía de UI-UX	9	12

Retomando el análisis de la Tabla 2. y con miras a continuar con el segundo objetivo, se realiza el conteo por categoría en la Tabla 3. Acá se puede observar la distribución de los *tickets* según su clasificación. A continuación, se presentarán dos diagramas circulares que permiten visualizar la distribución de las tareas, permitiendo un análisis más detallado de los resultados obtenidos en el desarrollo de la práctica académica.

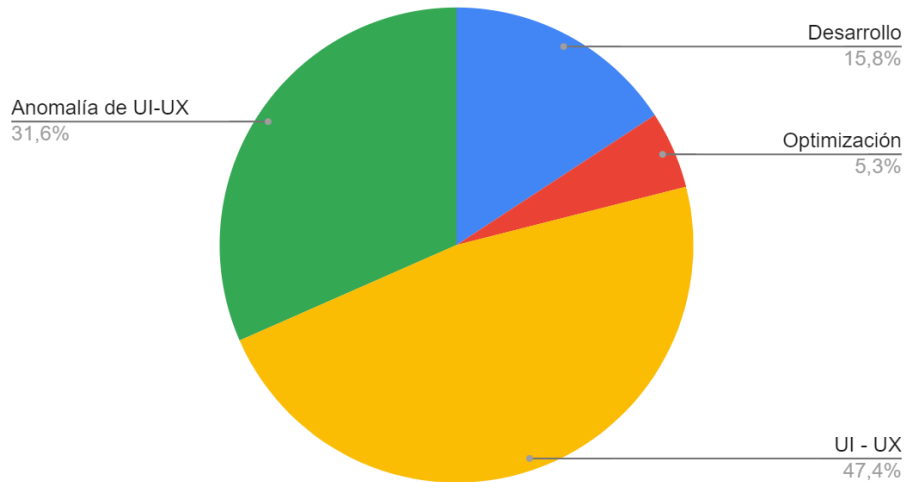
Imagen 5 Gráfico de cantidad de tickets por categoría



En primer lugar, al observar la imagen 5, se puede apreciar que una parte significativa de la práctica estuvo dedicada a la resolución de anomalías en el área de UI-UX, representando un 45% del total de los *tickets*. Este hallazgo revela que se puso un esfuerzo considerable en abordar y mejorar las experiencias de usuario existentes en el proyecto.

Imagen 6 Gráfico de puntos de historia por categoría

Puntos de historia por categoría



Por otro lado, al analizar la imagen 6, se obtiene una perspectiva más detallada de los puntos de historia asignados a cada categoría. En este sentido, se destaca que la mayor parte del trabajo se enfocó en la implementación de nuevas funcionalidades, representando un 47,4% de los puntos de historia asignados. Esto contrasta ligeramente con la distribución de los *tickets*, lo cual sugiere que, si bien se abordaron una cantidad considerable de anomalías de UI-UX, se asignó una mayor carga de trabajo a la incorporación de nuevas funcionalidades.

Continuando con el análisis, se puede observar que el 15% de los *tickets* totales corresponden a tareas de desarrollo, como se muestra en la imagen 5. Este resultado indica que se dedicó una proporción significativa de la práctica a la mejora del código o herramientas para los desarrolladores. Además, al considerar los puntos de historia asignados, se encuentra que el 15,8% estuvo relacionado con el desarrollo, lo que respalda la importancia de esta categoría en términos de complejidad y esfuerzo requerido.

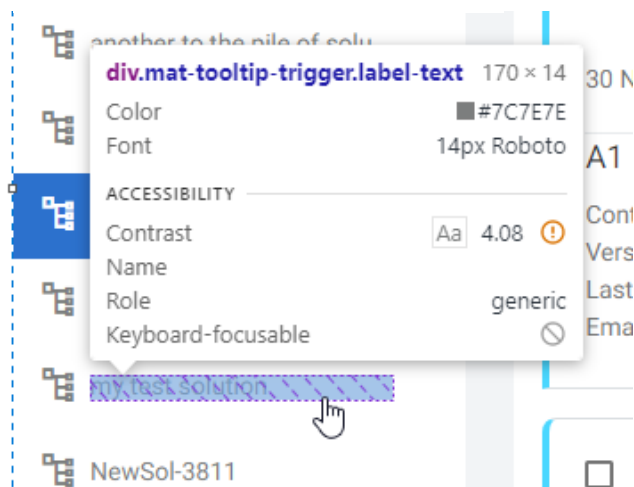
Asimismo, se identifica que el 10% de los *tickets* totales se enfocaron en la categoría de optimización, como se evidencia la imagen 5. Esto demuestra un esfuerzo considerable destinado a mejorar el rendimiento y la eficiencia del sistema. En cuanto a los puntos de

historia, se asignó un 5,3% a tareas de optimización, lo que indica una dedicación proporcional a esta categoría en relación con el resto.

En resumen, los gráficos brindan una visión clara de la distribución de los *tickets* y los puntos de historia en cada categoría. Estos resultados destacan la importancia de abordar las anomalías de UI-UX y la implementación de nuevas funcionalidades, con una dedicación adicional a las herramientas de desarrollo y optimización. Esta información permite evaluar el progreso y el impacto de la práctica académica en las áreas prioritarias del proyecto.

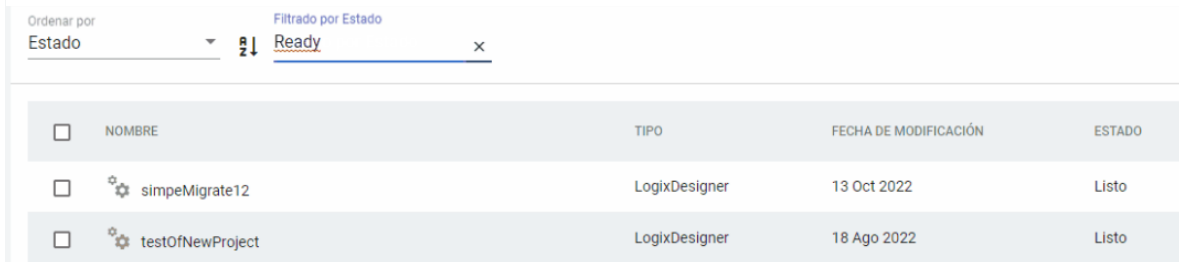
Con el propósito de complementar el análisis general y mostrar las contribuciones en diferentes aspectos del desarrollo del producto FTVault, se mostrarán imágenes de *tickets* seleccionados. Estas representan el trabajo realizado durante la práctica académica, brindando una muestra de las tareas efectuadas para mejorar la experiencia e interfaz de usuario.

Imagen 7 Anomalía de UI-UX (4233)



En la Imagen 7, se puede apreciar una particularidad relacionada con la interacción del usuario en el área de un componente visual del proyecto FTVault. Anteriormente, la funcionalidad de hacer clic solo estaba habilitada sobre el texto del componente, representada por un pequeño segmento, en lugar de extenderse a toda el área en general.

Imagen 8 Anomalía de UI-UX (3895) parte 1



<input type="checkbox"/>	NOMBRE	TIPO	FECHA DE MODIFICACIÓN	ESTADO
<input type="checkbox"/>	⚙️ simpeMigrate12	LogixDesigner	13 Oct 2022	Listo
<input type="checkbox"/>	⚙️ testOfNewProject	LogixDesigner	18 Ago 2022	Listo

Imagen 9 Anomalía de UI-UX (3895) parte 2





En las imágenes 8 y 9, se destaca un problema relacionado con la búsqueda del estado en el proyecto FTVault. Se encontró que, mientras la búsqueda en inglés funcionaba correctamente, la búsqueda en español no arrojaba resultados, a pesar de tener el idioma configurado correctamente. El inconveniente se debió al uso de comparaciones con valores estáticos, lo que limitaba la búsqueda y requería buscar el estado en su traducción adecuada. Se implementaron mejoras para permitir una búsqueda efectiva en los diferentes idiomas soportados por el producto (ES-EN-IT-FR-CZ-CH), mejorando la usabilidad y accesibilidad del sistema para los usuarios que usan estos idiomas.

Imagen 10 Optimización - estado inicial (4157)

<input type="checkbox"/> solutions	/api/idh/vaults/42a363184...	OPTIONS	204	preflight
<input type="checkbox"/> solutions	/api/idh/vaults/42a363184...	GET + Preflight	200	xhr
<input type="checkbox"/> solutions	/api/idh/vaults/96602f155...	OPTIONS	204	preflight
<input type="checkbox"/> solutions	/api/idh/vaults/96602f155...	GET + Preflight	200	xhr
<input type="checkbox"/> solutions	/api/idh/vaults/a406e3a5b...	OPTIONS	204	preflight
<input type="checkbox"/> solutions	/api/idh/vaults/a406e3a5b...	GET + Preflight	200	xhr
<input type="checkbox"/> solutions	/api/idh/vaults/dab7a0450...	OPTIONS	204	preflight
<input type="checkbox"/> solutions	/api/idh/vaults/dab7a0450...	GET + Preflight	200	xhr

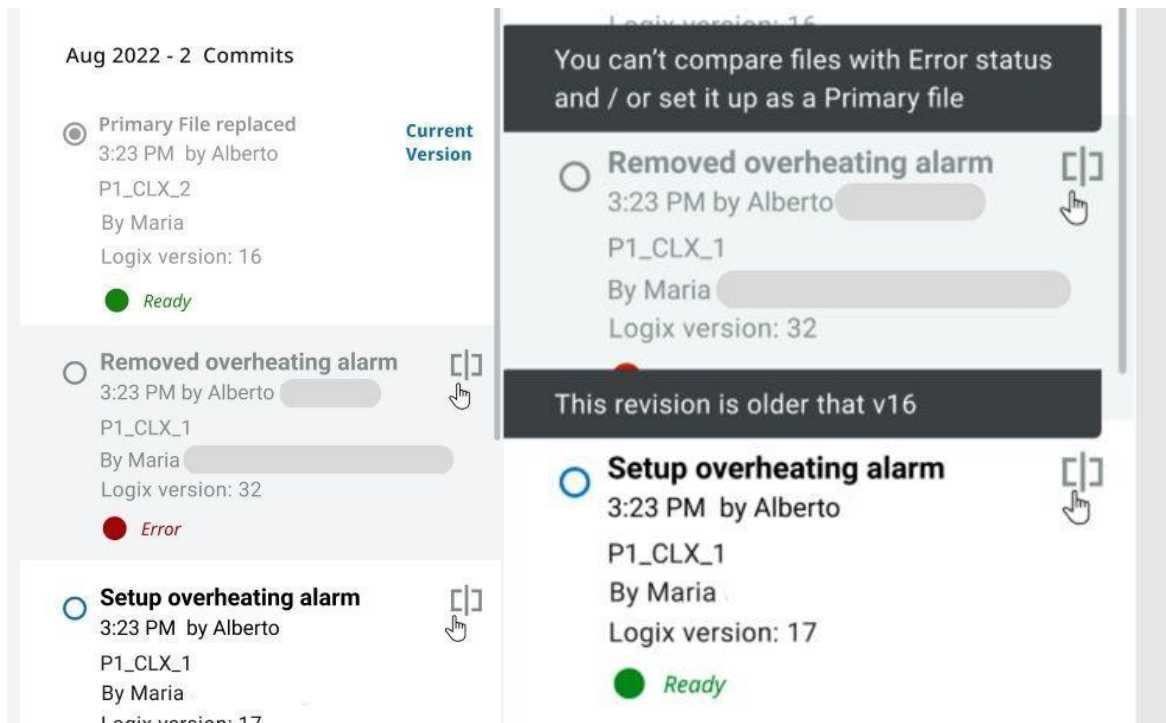
Imagen 11 Optimización - estado final (4157)

<input type="checkbox"/> metrics	200	xhr	sidebar-vault.c...	714 B	410 ms
<input type="checkbox"/> solutions	204	preflight	Preflight 	0 B	139 ms
<input type="checkbox"/> folders&thumbnail=true	204	preflight	Preflight 	0 B	148 ms

Ahora bien, en la etiqueta de optimización se encuentra una tarea interesante, no solo por el impacto hacia el producto, sino también por el aprendizaje obtenido. En la imagen 10 se puede observar que el reto encontrado fue la sobrecarga de una misma petición (*solutions*) al momento de cargar la página principal. Esta clase de sobrecarga perjudica el rendimiento del producto. Es por esto que resultó necesario tomar acción sobre esta y reducirla a una sola petición, como se puede ver en la imagen 11.

Continuando con otra categoría, algunas historias realizadas durante la práctica se enfocaron en la inserción de *tooltips* o mensajes que brindan al usuario una mayor información del estado que se tiene dentro del componente. Para este caso, el botón sobre el cual se tiene enfocado el *mouse* en la imagen 12 permite hacer la comparación entre dos textos o archivos. Debido a que existen casos estipulados en los requisitos funcionales en los que no se pueden comparar, es de importancia que el usuario tenga la razón y/o el contexto al intentar realizar esta acción y no sea posible, siendo esto logrado con los mensajes emergentes.

Imagen 12 Tarea de UI-UX (3863)



Siguiendo el orden que se lleva y centrando la atención en el tercer objetivo específico, la Imagen 12 muestra los resultados iniciales de las pruebas unitarias en el proyecto FTVault, con 1820 pruebas ejecutadas satisfactoriamente y una cobertura de declaraciones del 81.28% y del 81.6% en líneas de código. En la Imagen 13, los resultados finales revelan un progreso significativo con 2029 pruebas ejecutadas y una cobertura de declaraciones del 80.84% y del 81.08% en líneas de código.

Imagen 13 Resultados tests unitarios al inicio de la práctica

```
Chrome Headless 114.0.5735.199 (Windows 10): Executed 1820 of 1829 (skipped 9) SUCCESS (40.093 secs / 40.951 secs)
TOTAL: 1820 SUCCESS

===== Coverage summary =====
Statements : 81.28% ( 10369/12757 )
Branches   : 65.01% ( 5247/8070 )
Functions  : 78.09% ( 3001/3843 )
Lines      : 81.6% ( 10139/12424 )
=====
```

Imagen 14 Resultados tests unitarios al final de la práctica

```
Chrome Headless 114.0.5735.199 (Windows 10): Executed 2029 of 2040 (skipped 11) SUCCESS (41.188 secs / 31.655 secs)
TOTAL: 2029 SUCCESS

===== Coverage summary =====
Statements : 80.84% ( 11250/13915 )
Branches   : 63.86% ( 5770/9034 )
Functions  : 77.8% ( 3278/4213 )
Lines      : 81.08% ( 11033/13606 )
=====
```

Es relevante destacar que, a lo largo de la práctica, la cobertura de pruebas se mantuvo constantemente por encima del 80%, lo cual era la meta establecida para esta medida. Este logro resalta el compromiso del equipo con la calidad del producto y la importancia dada a las pruebas como parte fundamental del proceso de desarrollo. Los resultados obtenidos en las pruebas son el reflejo del esfuerzo colaborativo del equipo en su conjunto, demostrando el impacto positivo que la práctica tuvo en el proceso de garantizar un producto FTVault más robusto y confiable.

En resumen, la práctica académica fue un paso importante para identificar y priorizar oportunidades de mejora en el desarrollo *frontend* de FTVault, con el objetivo de brindar a los clientes de Rockwell Automation una experiencia de usuario más satisfactoria. Durante la implementación se resolvieron las tareas identificadas, se desarrollaron e implementaron nuevas funcionalidades y mejoras, que aportan mayor valor a los usuarios. Asimismo, se ha trabajado de manera rigurosa en la documentación y pruebas necesarias para asegurar la calidad del producto, siguiendo la metodología establecida por la empresa y cumpliendo con los requisitos de cada tarea.

Como resultado, se han solucionado anomalías identificadas, mejorando la experiencia de usuario y logrando avances significativos en la cobertura de pruebas, lo que ha fortalecido la calidad del producto FTVault. Estos esfuerzos conjuntos han permitido avanzar hacia la consecución de los objetivos específicos planteados, contribuyendo así al desarrollo exitoso de esta importante herramienta en el ámbito de la automatización industrial.

Conclusiones y trabajo futuro

Conclusiones

El proyecto en colaboración con el equipo de desarrollo *frontend* en el producto FTVault ha alcanzado con éxito los objetivos propuestos, evidenciando un enfoque efectivo en la mejora de la experiencia e interfaz de usuario, así como en la optimización del producto. Mediante la implementación de metodologías ágiles, como Scrum, se logró una gestión eficiente de las tareas y la priorización de oportunidades de mejora.

En primer lugar, la identificación y resolución de anomalías en el área de UI-UX ha sido una prioridad, lo que ha contribuido significativamente a fortalecer la calidad del producto y a satisfacer las necesidades de los usuarios. Las tareas de desarrollo e implementación de nuevas funcionalidades y mejoras, en línea con los objetivos específicos, han enriquecido el producto, aportando un mayor valor a los usuarios y mejorando su experiencia.

Además, el compromiso del equipo con la calidad del producto se ha reflejado en la aplicación de pruebas unitarias, manteniendo la cobertura de más del 80%, superando la meta establecida. La dedicación en la documentación y pruebas ha asegurado la excelencia del producto FTVault, con un enfoque en la eficiencia y el rendimiento, y garantizando la satisfacción del *Product Owner* y de los *stakeholders* o interesados involucrados.

Además de los logros mencionados, se destaca el resultado positivo de la optimización en el proyecto FTVault. Mediante la reducción de la cantidad de peticiones, especialmente en el área de sobrecarga identificada en la imagen 8, se ha logrado mejorar significativamente el rendimiento general del producto. Al implementar la mejora y reducir la sobrecarga a una sola petición, como se muestra en la imagen 9, se ha optimizado el tiempo de carga y respuesta del sistema, lo que se traduce en una experiencia más ágil y satisfactoria para los usuarios.

En conclusión, el proyecto ha sido una experiencia enriquecedora y exitosa, donde la implementación de metodologías ágiles y el trabajo colaborativo han sido fundamentales para lograr los resultados esperados. El enfoque en la experiencia de usuario, la incorporación

de nuevas funcionalidades y la optimización del producto FTVault han contribuido a su evolución y a su posicionamiento en el mercado. El equipo ha demostrado su capacidad y compromiso, alcanzando los objetivos propuestos y asegurando la satisfacción de los clientes.

Trabajo futuro

Como trabajo futuro, se puede considerar la exploración y desarrollo de metodologías específicas para abordar cada tipo de categoría encontrada en la clasificación de *tickets*. Por ejemplo, para las anomalías de UI-UX, se podría investigar y establecer mejores prácticas de diseño centradas en la usabilidad y la experiencia del usuario, con el objetivo de reducir la incidencia de este tipo de problemas. Para las tareas de desarrollo, se podría considerar la implementación de listas de chequeo de los principales factores a tener en cuenta antes de realizar un cambio en código.

Asimismo, se podría explorar y aprender sobre nuevas herramientas de análisis de código estático y dinámico, así como herramientas de monitoreo de rendimiento, para identificar áreas de mejora y optimización en el código y en el rendimiento del producto FTVault. Esto permitiría detectar de manera proactiva posibles problemas y oportunidades de mejora, garantizando una mayor calidad y rendimiento del producto.

En resumen, como trabajo futuro, se puede explorar la creación de metodologías específicas para abordar cada categoría de *tickets*, incorporar los resultados de las herramientas de análisis y monitoreo, y enfocarse en prácticas de diseño accesibles e inclusivas. Estas iniciativas contribuirían a mejorar aún más el producto FTVault y a satisfacer las necesidades y expectativas de los usuarios y los *stakeholders*.

Referencias

- [1] Elmhurst Public Library (s.f.) UI/UX Basics. Recuperado de https://elmhurstpubliclibrary.org/lib/wp-content/uploads/UIUXBasics_Handout_021918jj.pdf
- [2] Mozilla. (2021). Front-end Web Developer. Recuperado de https://developer.mozilla.org/es/docs/Learn/Front-end_web_developer
- [3] Atlassian. (s.f.). La metodología ágil. Recuperado de <https://www.atlassian.com/es/agile#:~:text=La%20metodolog%C3%ADa%20%C3%A1gil%20es%20un,r%C3%A1pida%20y%20con%20menos%20molestias>
- [4] MDN Web Docs. (2021). Web frameworks. Recuperado de https://developer.mozilla.org/es/docs/Learn/Server-side/First_steps/Web_frameworks
- [5] The Connected Enterprise (s.f.) Rockwell Automation. Recuperado de <https://www.rockwellautomation.com/es-co/capabilities/connected-enterprise.html>
- [6] Frost, B. (2016). Atomic Design. Brad Frost LLC. Recuperado de <https://bradfrost.com/blog/post/atomic-web-design/>
- [7] Martin, R. C. (2008). Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall.
- [8] Norman, D., & Nielsen, J. (2010). The definition of usability. Recuperado de <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>
- [9] The Collaborative Interface Design Tool. Figma. (s.f.) Recuperado de <https://www.figma.com/>
- [10] Fernández Carrasco, O. M., García León, D., & Beltrán Benavides, A. (1995). Un enfoque actual sobre la calidad del software. ACIMED, 3(3), 40-42. Recuperado de

http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94351995000300005&lng=es&tlng=es

- [11] IEEE Computer Society. (2011). "IEEE Recommended Practice for Software Requirements Specifications" (IEEE Std 830-2011).
- [12] Cohn, M. (2005). Agile Estimating and Planning. Prentice Hall.
- [13] Karttunen, J. (2016). Unit testing of AngularJS: A look into writing tests for web application (Bachelor's thesis). Degree Programme in Software Engineering, Technology, Communication and Transport. Recuperado de https://www.theseus.fi/bitstream/handle/10024/112102/karttunen_joel.pdf