

Desarrollo de una Plataforma de Gestión de Proyectos Informáticos basada en Diseño Centrado en el Usuario

*TRABAJO DE FIN DE MÁSTER
MÁSTER UNIVERSITARIO EN INGENIERÍA INFORMÁTICA*



**VNiVERSiDAD
D SALAMANCA**

CAMPUS DE EXCELENCIA INTERNACIONAL

AUTOR

DAVID MONTAGUT PAMO

TUTOR

DR. ROBERTO THERÓN SÁNCHEZ

Resumen

A la hora de abordar la creación de un proyecto informático, nos podemos encontrar con diferentes metodologías: tradicionales, ágiles y el Diseño Centrado en el Usuario (DCU)... Esta última destaca por la mejora en la experiencia del usuario, diseñando mientras se tiene en cuenta las necesidades y deseos de los usuarios finales. Este enfoque no solo hace que el producto final resulte útil y agradable, sino que también ayuda a retener a los usuarios. Como desarrolladores, el DCU nos ayuda a mitigar errores al identificar problemas potenciales de forma temprana.

Sin embargo, a pesar de las ventajas que ofrece el DCU es una metodología que puede ofrecer diferentes enfoques que, agregado a su naturaleza cíclica, puede llegar a acentuar el replanteamiento de decisiones ya tomadas. Todo ello para el objetivo de beneficiar al usuario final. Para los inexpertos, seguir el DCU puede resultar complejo debido a la naturaleza iterativa y las diversas metodologías aplicadas en cada etapa del proceso.

Es en este contexto donde se vuelve útil una herramienta que permita reflejar cada paso de manera ordenada y clara. Además, es beneficioso contar con una guía que muestre, de forma concisa, las distintas opciones disponibles para realizar cada apartado del proceso.

La herramienta que se presenta en este proyecto tiene la finalidad de ayudar a los inexpertos a gestionar su primer proyecto utilizando el Diseño Centrado en el Usuario. Al mismo tiempo, pretende ser una herramienta útil para aquellos usuarios más experimentados, permitiéndoles organizar su proyecto de una forma más ordenada y eficaz.

Palabras clave

- Diseño Centrado en el Usuario (DCU)
- Metodologías de desarrollo de software
- Gestión de proyectos informáticos
- Experiencia de usuario (UX)
- Desarrollo web
- Framework Laravel
- Tabla Kanban
- Prototipos de software

Abstract

When approaching the creation of a software project, we encounter different methodologies: traditional, agile, and User-Centered Design (UCD). The latter stands out for enhancing user experience by designing with the needs and desires of end users in mind. This approach not only ensures the final product is useful and enjoyable but also aids in user retention. As developers, UCD helps us mitigate errors by identifying potential issues early on.

However, despite the advantages offered by UCD, it is a methodology that can present different approaches, compounded by its iterative nature, which can accentuate the reconsideration of previously made decisions—all with the aim of benefiting the end user. For those new to it, following UCD can be complex due to its iterative nature and the varied methodologies applied at each stage of the process.

It is in this context that a tool becomes useful, one that allows each step to be reflected in an organized and clear manner. Furthermore, having a guide that succinctly presents the different options available for each section of the process is beneficial.

The tool presented in this project aims to assist novices in managing their first project using User-Centered Design. Simultaneously, it aims to be a valuable tool for more experienced users, enabling them to organize their projects in a more structured and effective manner.

Keywords

- User-Centered Design (UCD)
- Software development methodologies
- Software project management
- User experience (UX)
- Project management tool
- Web development
- Laravel framework
- User interface (UI)
- Kanban board
- Software prototyping

D. Roberto Therón Sánchez profesor del Departamento de Informática y Automática de la Universidad de Salamanca.

HACE CONSTAR:

Que el trabajo titulado “Desarrollo de una Plataforma de Gestión de Proyectos Informáticos basada en Diseño Centrado en el Usuario” ha sido realizado por D. David Montagut Pamo, con DNI 70.938.908P y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Máster de la Titulación Máster Universitario en Ingeniería Informática de esta Universidad.

Y para que así conste a los efectos oportunos.

En Salamanca a 3 de septiembre de 2024

Contenido

Resumen.....	2
Palabras clave.....	2
Abstract	3
Keywords.....	3
Introducción.....	7
Contexto.....	7
Organización del documento.....	8
Objetivos del proyecto	9
Objetivos software	9
Objetivos del alumno	9
Conceptos teóricos.....	10
Diseño centrado en el usuario:	10
Etapas del DCU implementadas por IA.....	13
<i>Discovery</i>	13
<i>Concepting</i>	14
<i>Prototyping and User Testing</i>	15
Términos y conceptos empleados.....	16
Técnicas y herramientas.....	17
Herramientas tecnológicas:.....	17
Proceso seguido	19
Fase de Discovery 1:.....	20
Tendencias en la industria.....	20
Competencia.....	20
Definición de usuarios:	21
Perfiles de usuario.....	23
Entornos de uso.....	24
Fase de <i>Concepting</i> 1:.....	25
Fase de <i>Prototyping and Testing</i> 1:.....	29
Prototipo en Figma.....	29
Desarrollo de la aplicación:.....	35
Fase de <i>Discovery</i> 2:.....	41
Perfiles de Usuario.....	41
Segmentar los Usuarios	41
Fase de <i>Prototyping and Testing</i> 2:	42
Entrevista de concepto	45

Entrevista para comprobar los iconos usados en la aplicación.	50
Cómo probar la aplicación.	52
Evaluación heurística	53
Conclusiones y trabajo futuro.....	56
Anexo 1 Manual del programador	58
Estructura de ficheros	58
Pasos para hacer la instalación del proyecto	62
Actualizar el sistema	62
Instalar PHP y Docker	62
Crear Docker MariaDB.....	62
Instalar Composer	63
Clonar el repositorio o descargarlo	63
Instalar node.....	63
Instalar paquetes de php	63
Activar las extensiones en php.ini	63
Lanzar el proyecto	63
Anexo 2 Manual de usuario.....	64
Anexo 3 Documentación del proyecto generada por la propia aplicación.	74
Anexo 4 Recursos y videos.	75
Entrevistas prototipo Figma	75
Primera versión de la aplicación	75
Segunda versión de la aplicación	75
Tercera versión de la aplicación	75
Entrevista <i>Affordance</i>	75
Bibliografía:	76

Introducción

En esta memoria se presentará el proceso seguido para la realización de una aplicación web destinada a la gestión de la planificación y desarrollo de un proyecto informático, aplicando la metodología del Diseño Centrado en el Usuario (DCU). A lo largo del proceso, se ha intentado seguir esta metodología, adaptándola a las necesidades específicas del momento, con el objetivo de lograr un mayor acercamiento al objetivo final.

Contexto.

Como graduado del Grado en Ingeniería Informática por la Universidad de Salamanca y actual estudiante del Máster en Ingeniería Informática en la misma universidad, he tenido la oportunidad de cursar asignaturas relacionadas con el DCU en dos ocasiones: "Interacción Persona/Ordenador" y "Paradigmas Avanzados de la Interacción Persona/Ordenador". Ambas asignaturas, dirigidas por mi tutor de este proyecto, garantizaban una mirada experta para resolver dudas y ayudarme en la realización de un producto funcional. Además, mi experiencia previa en estas materias me ha proporcionado una base sólida para abordar este proyecto de manera satisfactoria.

En cuanto a mi experiencia profesional, aunque aún no he tenido la oportunidad de implementar directamente el DCU en proyectos laborales, mi trabajo hasta la fecha me ha familiarizado con la importancia de seguir directrices y estándares establecidos. Este proyecto representa una valiosa oportunidad para aplicar los principios del DCU, permitiéndome crecer como ingeniero informático al explorar y diseñar soluciones desde la perspectiva del usuario.

Es relevante mencionar que, durante mis estudios de grado, el uso de inteligencias artificiales no era una posibilidad. Sin embargo, en la realización de este proyecto, se ha planteado el uso de una de las herramientas más conocidas en el mercado: ChatGPT, tanto en su versión 3.5 como en la 4.0. Esta herramienta se utilizará como apoyo en el proceso de desarrollo, específicamente para complementar o sustituir fases del Diseño Centrado en el Usuario (DCU) y para facilitar las tareas de codificación y desarrollo. Es importante destacar que la IA no se integrará directamente en la herramienta final del proyecto. En su lugar, se empleará ChatGPT para mejorar la eficiencia y calidad de las tareas relacionadas con el diseño, la documentación y la programación. De esta manera, se aprovechan las capacidades de la IA para optimizar el flujo de trabajo sin que forme parte del producto final entregable.

Organización del documento.

Este documento se ha estructurado con el objetivo de guiar al lector a través del desarrollo y justificación del proyecto de manera clara y lógica. La organización del contenido sigue un enfoque progresivo, donde se abordan los fundamentos teóricos, se detallan las metodologías empleadas, y se discuten los resultados obtenidos, concluyendo con una reflexión sobre las lecciones aprendidas y posibles mejoras futuras.

1. Introducción

Se presenta el contexto y motivación del proyecto, así como la relevancia del Diseño Centrado en el Usuario (DCU) en el desarrollo de la herramienta propuesta. Además, se describe la organización del documento para facilitar la navegación del lector.

2. Objetivos del Proyecto

Este apartado desglosa los objetivos del proyecto en dos vertientes: los objetivos técnicos relacionados con el desarrollo del software y los objetivos personales enfocados en el aprendizaje y desarrollo profesional.

3. Conceptos Teóricos

Se ofrece una síntesis de los conceptos clave relacionados con el DCU, su importancia y aplicación en el ámbito de la gestión de proyectos informáticos.

4. Técnicas y Herramientas

Se detallan las herramientas tecnológicas y metodologías empleadas en el desarrollo de la aplicación, destacando cómo cada una contribuye al cumplimiento de los objetivos del proyecto.

5. Proceso Seguido

En este apartado se describe el proceso de desarrollo del proyecto, dividiéndolo en fases que siguen el ciclo iterativo del DCU. Cada fase incluye una explicación de las actividades realizadas, las decisiones tomadas y los resultados obtenidos.

6. Conclusiones y Trabajo Futuro

Se reflexiona sobre los resultados del proyecto, evaluando el grado en que se han alcanzado los objetivos propuestos. Además, se sugieren posibles líneas de mejora y futuras ampliaciones de la herramienta desarrollada.

7. Anexos

Se incluyen documentos complementarios como manuales de uso, guías de instalación, y recursos adicionales que apoyan la implementación y utilización de la herramienta desarrollada.

8. Bibliografía

Finalmente, se recoge una lista de las referencias bibliográficas utilizadas a lo largo del proyecto, las cuales fundamentan teórica y metodológicamente el desarrollo del mismo.

Objetivos del proyecto

En este apartado se presentan los objetivos del proyecto, divididos en dos categorías principales: objetivos de software, que se centran en el desarrollo y funcionalidades de la herramienta, y objetivos personales, que están orientados a las metas de aprendizaje y desarrollo profesional del alumno.

Objetivos software

El objetivo principal de este proyecto es desarrollar una herramienta accesible desde cualquier dispositivo con acceso a internet para gestionar un proyecto informático que siga la metodología del Diseño Centrado en el Usuario (DCU).

Este objetivo principal se desglosa en los siguientes subobjetivos:

- **Estudio y Revisión:** Investigar y revisar las mejores prácticas en DCU y gestión de proyectos informáticos.
- **Análisis de Requisitos:** Identificar y analizar los requisitos que los equipos de desarrollo necesitan para gestionar un proyecto.
- **Diseño de Interfaz:** Crear una interfaz de usuario intuitiva, fácil de usar y eficiente.
- **Documentación y Transparencia:** Proporcionar a los usuarios una visión clara del proceso de desarrollo de sus proyectos, así como facilitar la generación de documentación de este.

Objetivos del alumno

Uno de los principales motivos para elegir este proyecto como trabajo de fin de máster es la oportunidad de mejorar mis habilidades en el desarrollo web, tanto en el *frontend* como en el *backend*. Para ello, me propongo los siguientes objetivos específicos:

- **Uso de PHP y Laravel:** Utilizar PHP como lenguaje de backend, un lenguaje con el que tengo experiencia tanto en mi Trabajo de Fin de Grado (TFG) como en mi trabajo actual. Sin embargo, esta vez emplearé el framework Laravel, con el cual no tengo experiencia previa, para ampliar mis conocimientos y habilidades.
- **Implementación en JavaScript:** Primar el uso de JavaScript en el desarrollo del *frontend*. Esto permitirá reducir la ejecución de código del lado del servidor y minimizar el refresco constante de la herramienta al usar cualquier funcionalidad.
- **Uso de Inteligencia Artificial:** Explorar y aplicar el uso de herramientas de inteligencia artificial, como ChatGPT, para asistir en el proceso de desarrollo. Esto incluye la utilización de la IA para generar ideas, optimizar código y mejorar la experiencia de usuario.

Conceptos teóricos.

El Diseño Centrado en el Usuario (DCU) es una metodología ampliamente reconocida y utilizada en el ámbito del desarrollo de software e interacción persona-computadora. Esta metodología, que tiene sus raíces en los años 80, se ha consolidado a lo largo de las décadas como una práctica esencial para garantizar que los productos y servicios tecnológicos sean accesibles, eficientes y agradables para los usuarios finales.

Dado que el DCU cuenta con una extensa trayectoria y una abundante literatura académica y profesional que aborda sus principios, técnicas y beneficios, es razonable asumir que los destinatarios de este Trabajo de Fin de Máster (TFM) ya poseen un conocimiento considerable sobre el tema. En consecuencia, este apartado se centrará en proporcionar un resumen conciso de los conceptos fundamentales del DCU. Esta aproximación permitirá contextualizar adecuadamente el proyecto sin redundar en detalles ya bien conocidos por el público especializado.

Diseño centrado en el usuario:

[1] El Diseño Centrado en el Usuario (DCU), también conocido como "User-centred design", es una filosofía de diseño que coloca al usuario y sus necesidades en el centro del proceso de desarrollo de productos. Esta metodología implica la participación activa de los usuarios en todas las fases del proyecto y se apoya en diversas técnicas de investigación y diseño para garantizar que los productos resultantes sean fáciles de usar y funcionales.

Principios del DCU según la norma ISO 9241-210:2019:

- Conocimiento explícito de los usuarios: El diseño se basa en una comprensión profunda de los usuarios, sus tareas y entornos.
- Participación del usuario: Los usuarios están implicados en el diseño y desarrollo del producto.
- Evaluación centrada en el usuario: El diseño es impulsado por evaluaciones centradas en el usuario.
- Proceso iterativo: El diseño es iterativo, permitiendo mejoras continuas basadas en la retroalimentación.
- Experiencia del usuario completa: El diseño aborda la totalidad de la experiencia del usuario.
- Perspectiva multidisciplinar: El equipo de diseño incluye diferentes disciplinas para abordar diversos aspectos del diseño.

Etapas del DCU:

- Entender el contexto de uso: Identificar a los usuarios, sus necesidades y cómo utilizarán el producto.
- Especificar los requisitos: Determinar los requisitos tanto de la empresa diseñadora como de los usuarios.
- Diseñar las soluciones y evaluar los resultados: Desarrollar el producto basado en los requisitos especificados y verificar si el producto cumple con los objetivos establecidos mediante evaluaciones continuas.

Aplicación del DCU en productos digitales:

En el ámbito de productos digitales como sitios web, aplicaciones y productos multimedia, el DCU se enfoca en aspectos como la accesibilidad, legibilidad, visibilidad y el lenguaje utilizado. La iteración es clave en este proceso, permitiendo la mejora continua del producto basándose en los conocimientos adquiridos en cada ciclo de diseño.

Este enfoque integral e iterativo del DCU asegura que los productos sean no solo funcionales, sino también accesibles y centrados en las necesidades reales de los usuarios, lo que resulta en una mejor experiencia de usuario y mayor satisfacción del cliente.

En la propia herramienta hay un apartado donde se facilitarán al usuario las diferentes etapas del DCU. Ese apartado se centrará en el siguiente ciclo:

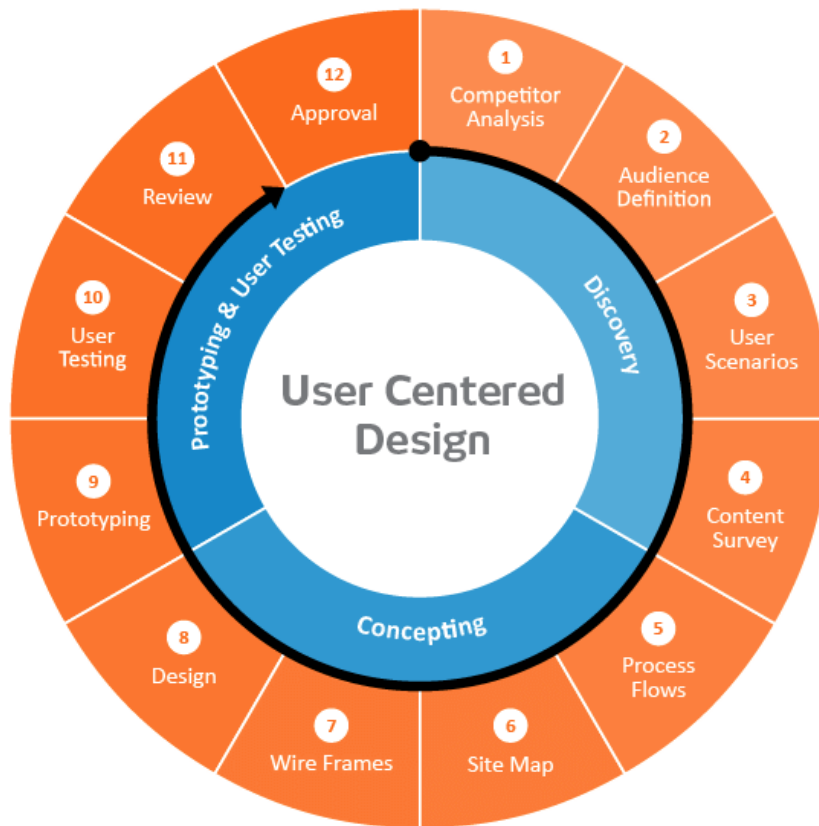


Figura 1 Ciclo del Diseño Centrado en el Usuario [2]

Etapas del DCU implementadas por IA

Se ha pedido a la inteligencia artificial que desarrolle el ciclo de la figura 1 para proporcionar breves descripciones sobre cada apartado y subapartado además de solicitar distintas metodologías para cada uno de los subapartados. Esta información será la que verán los usuarios mientras utilicen la aplicación.

Discovery

Descripción: Entender el mercado, los competidores y las necesidades de los usuarios para informar la dirección del proyecto.

- **Análisis de Competidores:**
 - **SWOT Analysis:**
 - "El análisis SWOT ayuda a identificar y evaluar las fuerzas y debilidades internas de la empresa, así como las oportunidades y amenazas del entorno."
 - **Benchmarking:**
 - "El benchmarking es una técnica que consiste en tomar como referencia los mejores aspectos de otras empresas o productos para mejorar los propios."
- **Definición de Audiencia:**
 - **Encuestas**
 - "Las encuestas permiten a los diseñadores obtener una idea más amplia de las necesidades y expectativas de los usuarios."
 - **Entrevistas:**
 - "Las entrevistas permiten obtener una comprensión profunda de las actitudes y comportamientos de los usuarios."
- **Escenmontaarios de Usuario:**
 - **Desarrollo de Personas:**
 - "Crear personas es una técnica que ayuda a representar a los diferentes tipos de usuarios y sus necesidades específicas."
 - **Historias de Usuario:**
 - "Las historias de usuario son útiles para capturar los requisitos del sistema desde la perspectiva del usuario."
- **Encuesta de Contenido:**
 - **Auditoría de Contenido:**
 - "La auditoría de contenido permite evaluar y mejorar el contenido existente para asegurar su relevancia y calidad."
 - **Inventario de Contenido:**
 - "El inventario de contenido es una herramienta esencial para gestionar y organizar todo el contenido existente de manera eficiente."

Concepting

Descripción: Desarrollar ideas y conceptos iniciales para resolver problemas identificados de los usuarios.

- **Flujos de Proceso:**
 - **Diagramas de Flujo:**
 - "Los diagramas de flujo son esenciales para visualizar y documentar los procesos y las interacciones dentro del sistema."
 - **Diagramas Swimlane:**
 - "Los diagramas swimlane son útiles para mostrar las responsabilidades y flujos de trabajo de diferentes roles en un proceso."
- **Mapa del Sitio:**
 - **Card Sorting:**
 - "El *card sorting* es una técnica que permite organizar el contenido en categorías de manera intuitiva para los usuarios."
 - "El *tree testing* ayuda a evaluar la estructura del sitio y asegurar que los usuarios pueden encontrar la información fácilmente."
- **Wireframes:**
 - **Esbozos:**
 - "Los esbozos permiten visualizar rápidamente ideas de diseño y explorar diferentes opciones de manera económica."
 - **Prototipos de Baja Fidelidad:**
 - "Los prototipos de baja fidelidad son útiles para probar y refinar el diseño antes de avanzar a etapas más detalladas."
- **Diseño:**
 - **Mood Boards:**
 - "Los *mood boards* son herramientas visuales que ayudan a establecer la dirección estética del diseño."
 - **Guías de Estilo:**
 - "Las guías de estilo son esenciales para mantener la consistencia y calidad del diseño a lo largo del proyecto."

Prototyping and User Testing

Descripción: Crear prototipos y probar con usuarios para validar y refinar soluciones.

- **Prototipado:**
 - **Prototipos Interactivos:**
 - "Los prototipos interactivos permiten probar y refinar las interacciones del usuario de manera efectiva."
 - **Prototipado en Papel:**
 - "El prototipado en papel es una técnica rápida y económica para probar y refinar ideas de diseño."
- **Pruebas con Usuarios:**
 - **Pruebas de Usabilidad:**
 - "Las pruebas de usabilidad son esenciales para identificar problemas y mejorar la experiencia del usuario."
 - **Pruebas A/B:**
 - "Las pruebas A/B permiten comparar diferentes versiones de un diseño y elegir la más efectiva."
- **Revisión:**
 - **Evaluación Heurística:**
 - "La evaluación heurística es una técnica útil para identificar problemas de usabilidad basados en principios establecidos."
 - **Crítica de Diseño:**
 - "La crítica de diseño es esencial para obtener retroalimentación constructiva y mejorar la calidad del diseño."
- **Aprobación:**
 - **Revisión con Stakeholders:**
 - "La revisión con *stakeholders* es crucial para obtener la aprobación y asegurar el éxito del proyecto."
 - **Aprobación Final:**
 - "La aprobación final es el paso necesario para avanzar del diseño a la implementación del proyecto."

Términos y conceptos empleados

Tabla Kanban:

Es una herramienta visual utilizada en la gestión de proyectos para organizar tareas en columnas que representan diferentes etapas de un flujo de trabajo, como "Por hacer", "En progreso" y "Completado". Las tareas se representan mediante tarjetas que se mueven entre columnas a medida que avanzan en el proceso.

Tag:

Es una etiqueta o palabra clave asignada a un elemento (como una tarjeta Kanban) para categorizar y organizar información dentro de un sistema. En tu proyecto, los "Tags" se utilizan para clasificar tarjetas Kanban, permitiendo un acceso más rápido y organizado a la información.

Evaluación Heurística:

Es una técnica de revisión en la que expertos evalúan la interfaz de usuario de una aplicación basándose en principios de usabilidad establecidos. El objetivo es identificar problemas que podrían afectar la experiencia del usuario.

Wireframes:

Son representaciones esquemáticas de una página web o aplicación, que muestran la disposición básica de los elementos de la interfaz y la estructura de navegación, sin centrarse en el diseño visual o los detalles de estilo.

Scrum:

Scrum es una metodología ágil utilizada para la gestión de proyectos que organiza el trabajo en ciclos cortos llamados "sprints", permitiendo iteraciones rápidas y la adaptación constante a los cambios.

Usabilidad:

Se refiere a la facilidad con la que los usuarios pueden interactuar con una interfaz o sistema. En un contexto de desarrollo de software, la usabilidad es crucial para asegurar que los usuarios puedan completar sus tareas de manera eficiente y satisfactoria.

Técnicas y herramientas

En esta sección se presentan las técnicas y herramientas empleadas para la elaboración de este proyecto.

Herramientas tecnológicas:



- **Docker:**

Docker se utilizó para lanzar la base de datos de forma cómoda, facilitando la creación y gestión de entornos de desarrollo aislados y consistentes. Al contener la base de datos en un contenedor, se garantiza la portabilidad y la independencia de la infraestructura subyacente, lo que permite replicar el entorno de producción en cualquier máquina sin conflictos de dependencia. Además, Docker facilita el versionado de configuraciones y la integración con herramientas de CI/CD (Integración Continua/Despliegue Continuo), optimizando el flujo de trabajo del desarrollo.



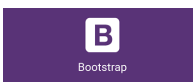
- **MariaDB:**

MariaDB, una base de datos relacional derivada de MySQL, se utilizó para almacenar y gestionar los datos de la aplicación. Su alta compatibilidad con MySQL y sus mejoras en rendimiento y seguridad la hacen una elección robusta para proyectos que requieren una base de datos confiable. MariaDB es conocida por su arquitectura escalable y su capacidad para manejar grandes volúmenes de datos, lo que asegura un rendimiento óptimo incluso bajo cargas de trabajo intensivas.



- **PHP:**

PHP, un lenguaje de programación ampliamente utilizado en el desarrollo web, fue seleccionado para el backend debido a su extensa trayectoria y su abundante documentación y recursos. PHP es especialmente adecuado para la construcción de aplicaciones web dinámicas y ofrece una amplia variedad de frameworks y bibliotecas que aceleran el desarrollo. Su integración con bases de datos relacionales como MariaDB facilita la creación de aplicaciones robustas y escalables.



- **Bootstrap:**

Bootstrap, un framework CSS, se empleó por su capacidad para crear rápidamente interfaces visualmente atractivas y responsive. Su colección de componentes predefinidos y su sistema de grillas facilitan el diseño de layouts adaptables a diferentes tamaños de pantalla, mejorando la experiencia del usuario en dispositivos móviles y de escritorio. Bootstrap también simplifica la incorporación de estilos consistentes y modernos en toda la aplicación.



- **JavaScript:**

JavaScript se utilizó para añadir interactividad y dinamismo a la aplicación web. Este lenguaje es fundamental para la creación de interfaces de usuario fluidas y responsivas, permitiendo la manipulación del DOM (Document Object Model) y la gestión de eventos en tiempo real. Además, JavaScript facilita la comunicación asíncrona con el backend mediante AJAX, mejorando la experiencia del usuario al reducir la necesidad de recargar la página.



- **PhpStorm:**

PhpStorm es un entorno de desarrollo integrado (IDE) optimizado para PHP, que ofrece potentes herramientas de desarrollo y depuración. Sus características incluyen la integración con sistemas de control de versiones, soporte para frameworks de PHP como Laravel, y una amplia variedad de plugins que mejoran la productividad del desarrollador. PhpStorm proporciona autocompletado de código, análisis estático y navegación avanzada, lo que facilita la escritura y el mantenimiento del código.



- **Laravel:**

Laravel, un framework de PHP para aplicaciones web, fue elegido por su extensa documentación y su ecosistema completo que incluye características como el ORM Eloquent, migraciones de base de datos y un sistema de enrutamiento intuitivo. Laravel simplifica tareas comunes como la autenticación, la gestión de sesiones y la validación de formularios, permitiendo a los desarrolladores centrarse en la lógica de negocio y en la creación de aplicaciones escalables y mantenibles.



-

Figma es una herramienta de diseño de interfaces y creación de prototipos que permite la colaboración en tiempo real. Esta herramienta se utilizó para diseñar y compartir prototipos interactivos, facilitando la revisión y prueba remota de los diseños. Figma soporta la creación de componentes reutilizables y estilos globales, asegurando la consistencia visual en todo el proyecto. Su capacidad para importar y exportar diferentes formatos de diseño optimiza la integración con otras herramientas de desarrollo.

Proceso seguido

En este apartado se abordará de forma esquemática los pasos seguidos para la realización del proyecto. Como se puede apreciar, el propio trabajo de fin de máster se ha abordado siguiendo el enfoque DCU.

1. Fase de *Discovery* 1:
 - 1.1. Tendencias en la industria.
 - 1.2. Análisis de la competencia.
 - 1.3. Definición de los usuarios objetivos.
 - 1.4. Entorno de uso.
2. Fase de *Concepting* 1:
 - 2.1. Diseños a papel.
3. Fase de *Prototyping and Testing* 1
 - 3.1. *Diseño* en Figma.
 - 3.2. Diseño de la base de datos.
 - 3.3. Entrevistas sobre el diseño en Figma.
 - 3.4. Codificación en Laravel del proyecto.
 - 3.5. Entrevista con el tutor.
4. Fase de *Discovery* 2
 - 4.1. Redefinir los usuarios objetivos
5. Fase de *Prototyping and Testing* 2
 - 5.1. Codificación del proyecto.
 - 5.2. Entrevista.
 - 5.3. Codificación del proyecto.
 - 5.4. Entrevista sobre el uso de los iconos en los botones.
 - 5.5. Codificación del proyecto.

Fase de Discovery 1:

En este punto de la memoria se detallarán las consideraciones previas que se tomaron para plantear la solución presentada con la aplicación:

Tendencias en la industria.

Se le pidió a la IA que me proporcionase tres tendencias sobre la industria de la gestión de proyectos:

- **Enfoque en Colaboración Remota:** La tendencia actual en la industria de gestión de proyectos informáticos destaca la importancia de las herramientas que facilitan la colaboración remota, especialmente dada la creciente adopción del trabajo a distancia.
- **Integración de Tecnologías Emergentes:** La integración de tecnologías emergentes como inteligencia artificial y análisis de datos en las plataformas de gestión de proyectos se está convirtiendo en una tendencia para mejorar la toma de decisiones y la eficiencia.
- **Énfasis en la Usabilidad:** Las plataformas exitosas priorizan la usabilidad, proporcionando interfaces intuitivas y experiencias de usuario fluidas.

Competencia.

Durante mi experiencia laboral he podido emplear varias aplicaciones para gestionar proyectos, como pueden ser Jira, Microsoft Teams o Odoo. Estas herramientas presentan formas de gestionar las tareas prioritariamente con el uso de tablas Kanban o cronogramas, chats con tu equipo y la posibilidad de subir archivos. Con esto en mente se le pidió a la IA la descripción de algunas de estas:

- **Jira:** [Jira](#), desarrollado por Atlassian, destaca por su potente seguimiento de problemas y personalización avanzada, pero puede tener una curva de aprendizaje pronunciada, especialmente para usuarios nuevos, debido a su amplia gama de funciones.
- **Odoo:** [Odoo](#) ofrece una integración completa de aplicaciones empresariales, brindando eficiencia en la gestión empresarial. Sin embargo, algunas empresas pueden encontrar su implementación inicial compleja debido a su amplio conjunto de características y personalización.
- **Microsoft Teams:** [Microsoft Teams](#) ofrece una sólida colaboración en tiempo real, pero algunos usuarios encuentran su interfaz abrumadora, especialmente para proyectos más complejos.
- **Trello:** [Trello](#) es conocido por su interfaz atractiva y fácil de usar, pero carece de ciertas funciones avanzadas que podrían ser cruciales para equipos más grandes y proyectos complejos.
- **Asana:** [Asana](#) se centra en la usabilidad y la colaboración remota, pero algunos usuarios han señalado que la gestión de proyectos a gran escala puede volverse complicada.

Definición de usuarios:

Para este apartado el uso de la IA me fue de gran utilidad ya que posee una gran fuente de datos sobre usuarios, tanto con características sociales como demográficas.

En un principio la aplicación tal y como yo la tenía pensada tendría el mismo público objetivo que mis competidoras, es decir, desarrolladores software, gerentes de proyectos y los clientes.

Descripción de este tipo de usuarios según la IA:

Desarrolladores de Software: Este grupo está compuesto principalmente por profesionales de entre 25 y 35 años, con una distribución de género equitativa. Suelen tener al menos un título universitario en informática o campos relacionados. Su ocupación es principalmente la programación y el desarrollo de software.

Características:

- Experiencia en programación superior a 5 años.
- Edades entre 30 y 40 años.
- Mayoritariamente masculino.
- Nivel de educación universitario en informática o campos relacionados.

Necesidades Comunes:

- Acceso a herramientas avanzadas de desarrollo.
- Colaboración eficiente con otros desarrolladores.
- Seguimiento de tareas y errores en tiempo real.

Objetivos Comunes:

- Desarrollar software de alta calidad.
- Cumplir con plazos de entrega.
- Mantener un código limpio y eficiente.

Gerentes de Proyecto: Los gerentes de proyecto son en su mayoría profesionales con edades comprendidas entre 30 y 50 años, y la distribución de género puede variar. Tienen una amplia gama de títulos de educación superior, desde administración hasta tecnología de la información. Su ocupación principal es la gestión de proyectos informáticos.

Características:

- Experiencia en gestión de proyectos de 3 a 7 años.
- Edades entre 30 y 50 años.
- Diversidad de género.
- Educación a nivel de máster o equivalente en gestión de proyectos.

Necesidades Comunes:

- Seguimiento del progreso del proyecto en tiempo real.
- Comunicación efectiva con equipos de desarrollo.
- Generación de informes personalizables.

Objetivos Comunes:

- Cumplir con los plazos del proyecto.
- Asignación eficiente de recursos.
- Satisfacer las expectativas del cliente.

Clientes y Stakeholders: Este grupo incluye personas de diferentes edades, desde 25 años en adelante. La distribución de género es equitativa. Sus niveles de educación varían, desde títulos universitarios hasta profesionales autodidactas. Estos usuarios pueden ocupar roles en diversas industrias y organizaciones y están interesados en el seguimiento y la colaboración en proyectos informáticos.

Características:

- Diversidad de edades, desde 25 años en adelante.
- Diversidad de género.
- Niveles de educación variados.

Necesidades Comunes:

- Visibilidad del progreso del proyecto.
- Comunicación efectiva con el equipo de desarrollo.
- Aportar comentarios y solicitudes.

Objetivos Comunes:

- Obtener un producto de software eficiente y dentro del presupuesto.
- Satisfacer las necesidades de su empresa o cliente.

Perfiles de usuario.

[3] Para poder imaginarse y empatizar con los tipos de usuarios mencionados previamente, es útil asignar estos roles a personas ficticias, dándoles nombres y objetivos específicos. Esta técnica, conocida como Diseño de Personas, fue popularizada por Alan Cooper y es ampliamente utilizada en el diseño centrado en el usuario.

El Diseño de Personas consiste en crear representaciones detalladas de diferentes segmentos de usuarios para guiar el proceso de diseño y desarrollo. Estas representaciones se basan en datos reales y ayudan a los equipos a comprender las necesidades, comportamientos y objetivos de sus usuarios. Al utilizar personas ficticias pero realistas, los diseñadores pueden empatizar mejor con los usuarios finales y crear soluciones más adecuadas y centradas en el usuario.

En este apartado, también nos apoyaremos en la inteligencia artificial para generar ideas y definir mejor estos perfiles de usuario.

Perfil 1: Desarrollador de Software

- **Nombre:** Juan Pérez
- **Edad:** 28 años
- **Género:** Masculino
- **Ubicación Geográfica:** Ciudad de México, México
- **Nivel de Educación:** Licenciatura en Ingeniería en Sistemas
- **Ocupación:** Desarrollador de Software en una empresa de tecnología
- **Características Personales:** Apasionado por la programación, altamente técnico, busca soluciones eficientes, trabajo en equipo, orientado a resultados.
- **Necesidades:** Herramientas de colaboración en tiempo real, seguimiento de tareas claras, acceso a documentación técnica.
- **Objetivos:** Entregar proyectos en plazo, mantener un código limpio y eficiente, mejorar sus habilidades técnicas.
- **Desafíos:** Coordinación eficiente con otros miembros del equipo, mantenerse actualizado con las últimas tecnologías.

Perfil 2: Gerente de Proyecto

- **Nombre:** María Rodríguez
- **Edad:** 35 años
- **Género:** Femenino
- **Ubicación Geográfica:** Madrid, España
- **Nivel de Educación:** Máster en Gestión de Proyectos
- **Ocupación:** Gerente de Proyecto en una empresa de consultoría
- **Características Personales:** Organizada, orientada a resultados, liderazgo, toma de decisiones.
- **Necesidades:** Seguimiento del progreso del proyecto, colaboración en equipo, generación de informes personalizables.
- **Objetivos:** Cumplir con los plazos del proyecto, asegurar la asignación eficiente de recursos, satisfacer las expectativas del cliente.
- **Desafíos:** Coordinar múltiples equipos, gestionar cambios en el alcance del proyecto.

Perfil 3: Cliente y Stakeholder

- **Nombre:** Carlos Sánchez
- **Edad:** 45 años
- **Género:** Masculino
- **Ubicación Geográfica:** Miami, Estados Unidos
- **Nivel de Educación:** Licenciatura en Administración de Empresas
- **Ocupación:** Director de TI en una empresa de logística
- **Características Personales:** Interesado en resultados comerciales, comunicación efectiva, expectativas claras.
- **Necesidades:** Visibilidad del progreso del proyecto, comunicación con el equipo de desarrollo, aportar comentarios.
- **Objetivos:** Obtener un producto de software eficiente y dentro del presupuesto, satisfacer las necesidades de su empresa.
- **Desafíos:** Comunicación efectiva con el equipo de desarrollo, definición de requisitos claros.

Entornos de uso

Por entorno de uso se entienden dos conceptos, tanto el lugar físico desde dónde se realiza la acción como la herramienta empleada para ello.

Lugares de uso:

Debido a los acontecimientos relacionados con la pandemia es un hecho que el trabajo en remoto está en auge, eso no quita que trabajo en la oficina no se siga realizando. Por ello caben destacar los siguientes lugares de uso para la aplicación final:

- **Despacho de Desarrolladores:** Los desarrolladores pueden utilizar la plataforma en sus lugares de trabajo, donde se espera que se concentren en tareas de desarrollo. Esto implica un entorno de oficina con acceso a computadoras de escritorio o portátiles.
- **Ubicación Remota:** Los clientes y desarrolladores pueden interactuar con la plataforma desde ubicaciones remotas, ya sea desde sus oficinas, hogares o cualquier lugar con conexión a Internet.

Tecnologías utilizadas:

- **Desarrolladores:** Utilizarán computadoras de alto rendimiento con diferentes sistemas operativos, por lo que la plataforma debe ser compatible con diversas configuraciones.
- **Gerentes de Proyecto:** Pueden acceder a la plataforma desde computadoras portátiles o dispositivos móviles mientras se desplazan, por lo que la interfaz debe ser adaptable y fácil de usar en diferentes dispositivos.
- **Clientes y Stakeholders:** Pueden utilizar la plataforma desde una variedad de dispositivos, incluyendo computadoras de escritorio, tabletas y teléfonos móviles, por lo que la plataforma debe ser compatible con múltiples plataformas.

Fase de *Concepting* 1:

En el apartado de conceptualización, se detalla cómo las conclusiones obtenidas en la fase anterior se transforman en un conjunto de ideas preliminares esbozadas en papel. Los puntos clave que deben reflejarse son:

- Un sistema para gestionar las tareas que se realicen durante el desarrollo del proyecto.
- Clasificación de las tareas según la fase del ciclo de desarrollo en la que se encuentren.
- Un sistema que permita la comunicación entre los miembros del equipo y la anotación de comentarios sobre las tareas.
- La herramienta debe ser colaborativa, permitiendo la participación de múltiples usuarios.
- Según los requerimientos del proyecto, la herramienta debe permitir la generación de documentación del proyecto.

Con estos elementos en mente, se realizaron bocetos en papel integrando estos conceptos:

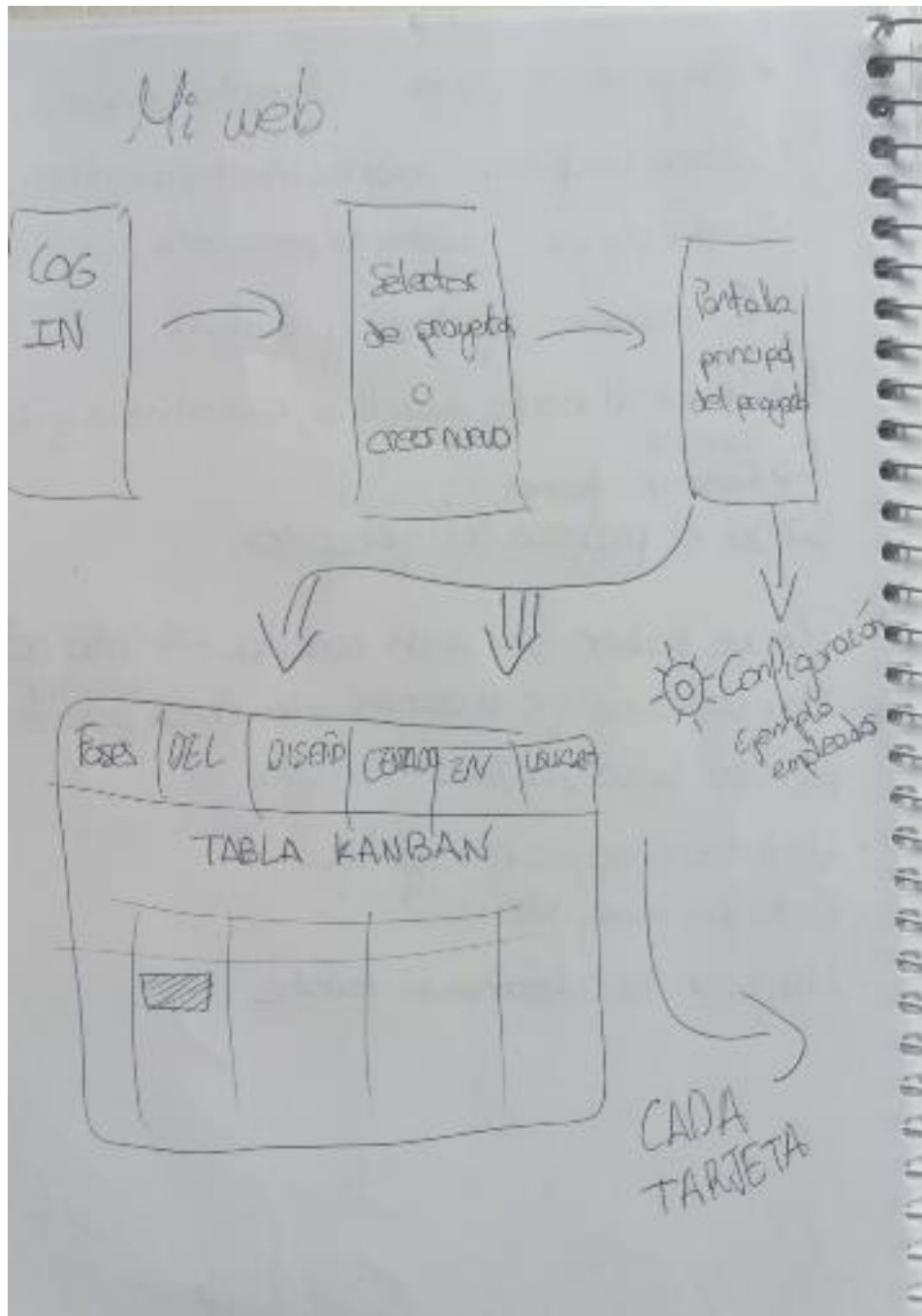


Figura 2 Boceto a papel 1

En la figura 2 se intenta reflejar el ciclo que seguirá el usuario al interactuar con la aplicación. Pasará por el LOGIN, accederá a sus proyectos o podrá crear uno nuevo. Una vez acceda a un proyecto podrá acceder a la tabla Kanban de ese proyecto y allí podrá ver una tabla Kanban diferente en función de en qué fase del proyecto esté.

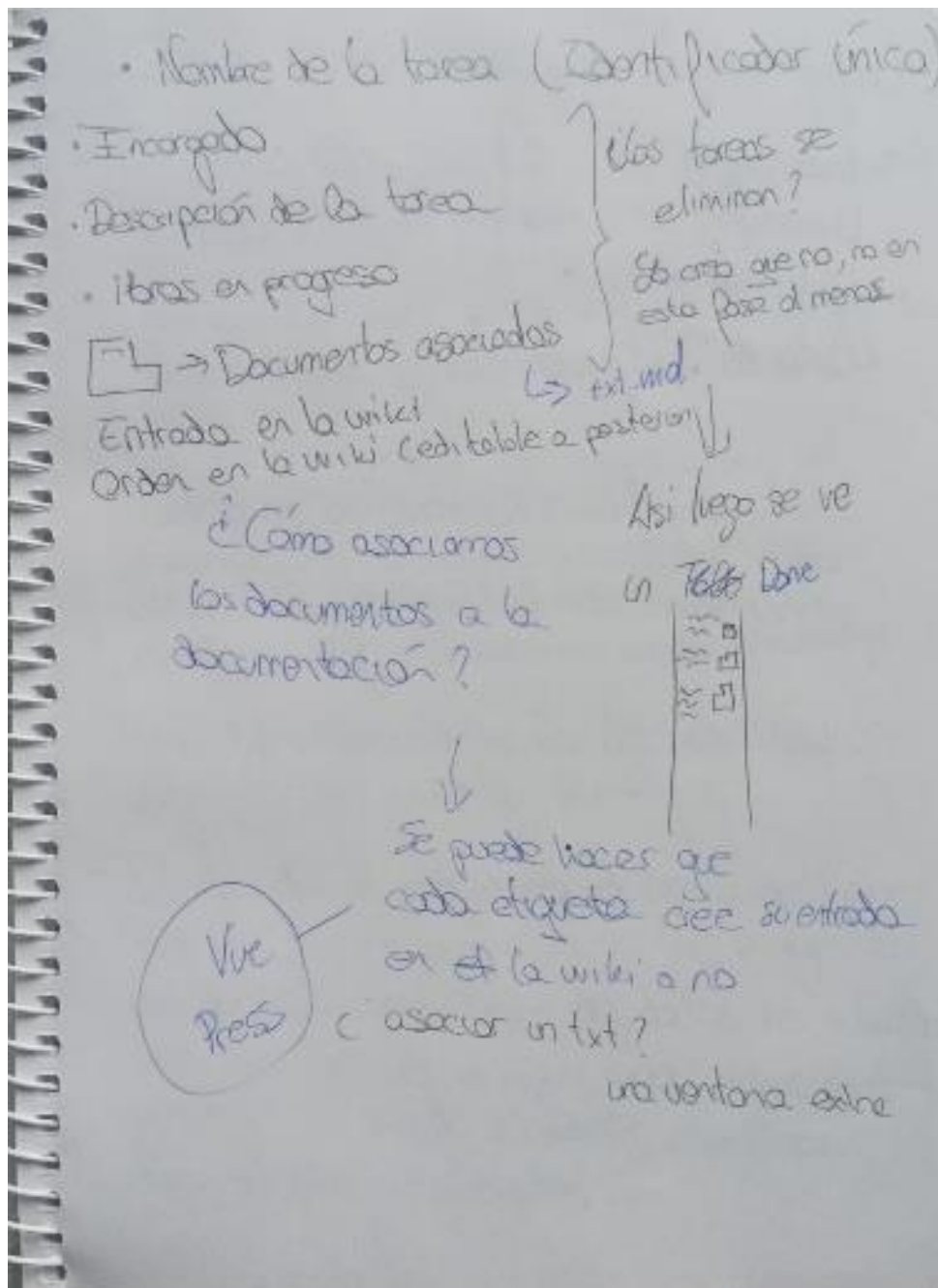
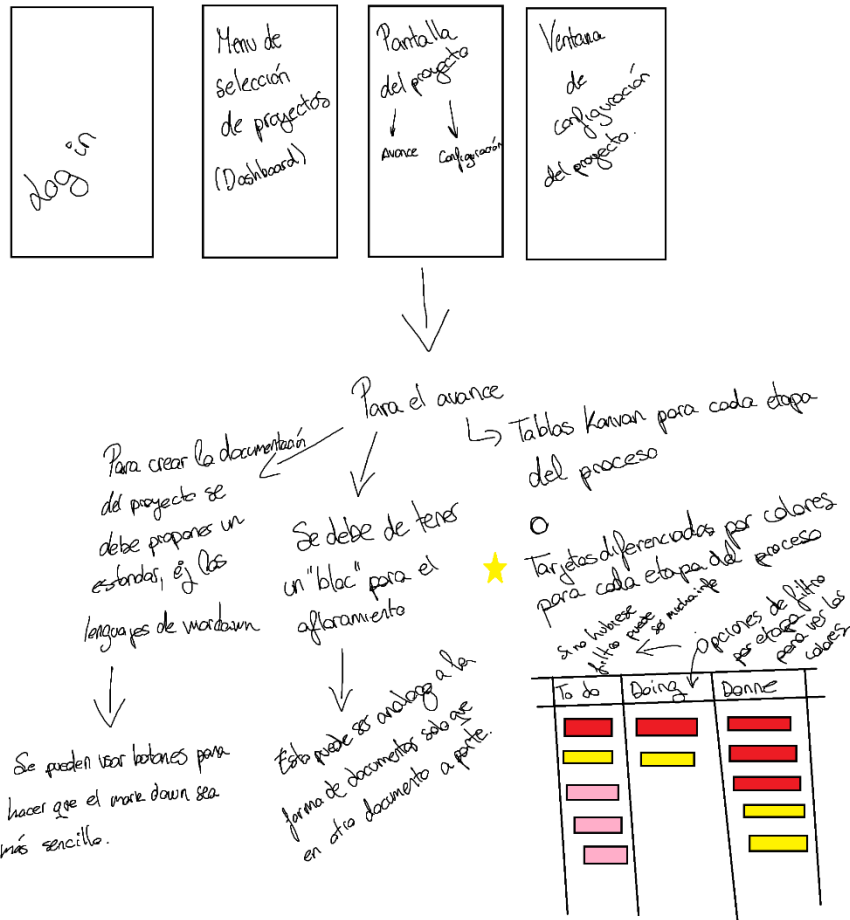


Figura 3 Boceto a papel 2

En estas anotaciones (ver figura 3) se refleja que tendrá cada tarea que forme la tabla Kanban, destacar que cada tarea podrá tener documentos asociados y la posibilidad de crear la documentación asociada a esa tarea desde la propia tarea.

Inicialmente, se plantea que todas las tareas puedan generar documentación, la cual se archivará en una Wiki. Para esta funcionalidad, se considera utilizar tecnología como [Vue Press](#)



- Será necesario poder tener diferentes permisos de gestión del proyecto
- Las tarjetas deberán llevar asociado un almacenamiento por si es necesario guardar elementos en el interior.
- Para crear la documentación se podría tener
 - 1 .md por etapa
 - 1 .md por tarjeta.

Figura 4 Boceto pasado a limpio

Una vez que las ideas iniciales (ver figura 4) han sido plasmadas y analizadas, se han llegado a las siguientes conclusiones para la implementación de la herramienta:

- **Generación de Documentación:** Se propone crear un estándar utilizando el lenguaje Markdown y desarrollar un editor incorporado en la aplicación para facilitar la creación y el formato de la documentación del proyecto.
- **Anotaciones durante el Proyecto:** Es crucial permitir la aparición y registro de anotaciones en cualquier momento durante el proceso del proyecto.
- **Tabla Kanban Unificada:** En lugar de mantener varias tablas Kanban para cada fase del DCU, se optará por una única tabla Kanban con la capacidad de filtrar tareas según la fase del proyecto. Esto simplificará la gestión y visualización de las tareas.
- **Permisos de Usuarios:** La herramienta debe incorporar un sistema de permisos de usuario diferenciados para garantizar un control adecuado de acceso y edición.

- **Archivos en Tarjetas:** Las tarjetas en la tabla Kanban deben permitir el almacenamiento de archivos en su interior para facilitar la organización y acceso a los recursos asociados a cada tarea.

Con estas conclusiones en mente, se puede proceder a la siguiente fase del proyecto.

Fase de *Prototyping and Testing 1:*

Prototipo en Figma

Tras haber plasmado las ideas iniciales en papel, se mantuvo una reunión con el tutor para evaluar si la herramienta propuesta cumplía con las expectativas. Tras su aprobación, se procedió a la creación del prototipo digital utilizando Figma.

Puedes acceder al prototipo a través del siguiente enlace: [Prototipo en Figma](#).

Usuario: ucdprojectsmanager@gmail.com

Contraseña: QaZwSxEdc123)=.

En caso de que haya algún problema con el enlace, el prototipo también está disponible en formato PDF, adjunto en la misma carpeta que el TFM.

Aquí un breve resumen de lo que se mostrará en el prototipo:

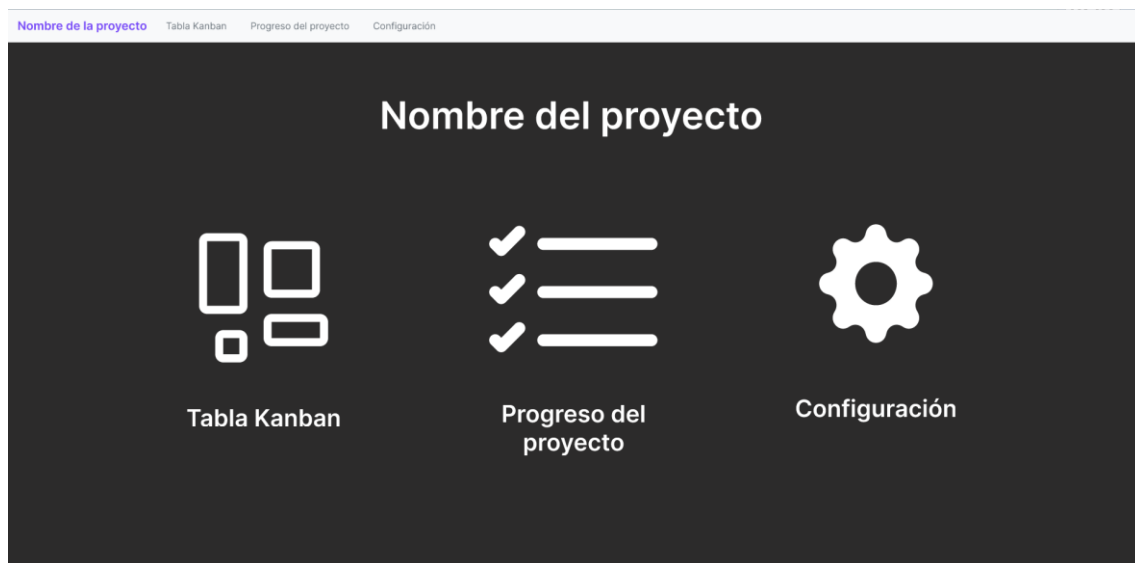


Figura 5 Menú principal

En la figura 5 se aprecia el menú principal del proyecto donde podremos navegar entre las distintas opciones Tabla Kanban (figura 6), Progreso del proyecto (figura 9) o la Configuración (Figura 10)

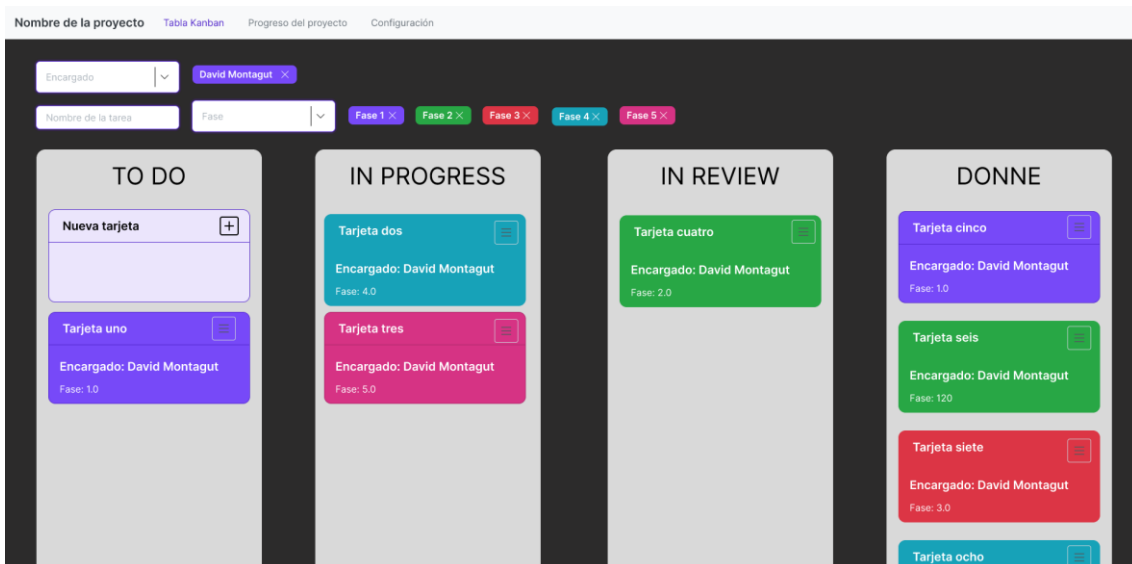


Figura 6 Tabla Kanban

En la figura 6 se muestra una tabla Kanban con tarjetas/tareas diferenciadas por colores en función de su fase dentro del proyecto. También se puede ver un filtro en la parte superior y la posibilidad de acceder a cada tarea desde un botón. Esta interacción de muestra en la figura 7.

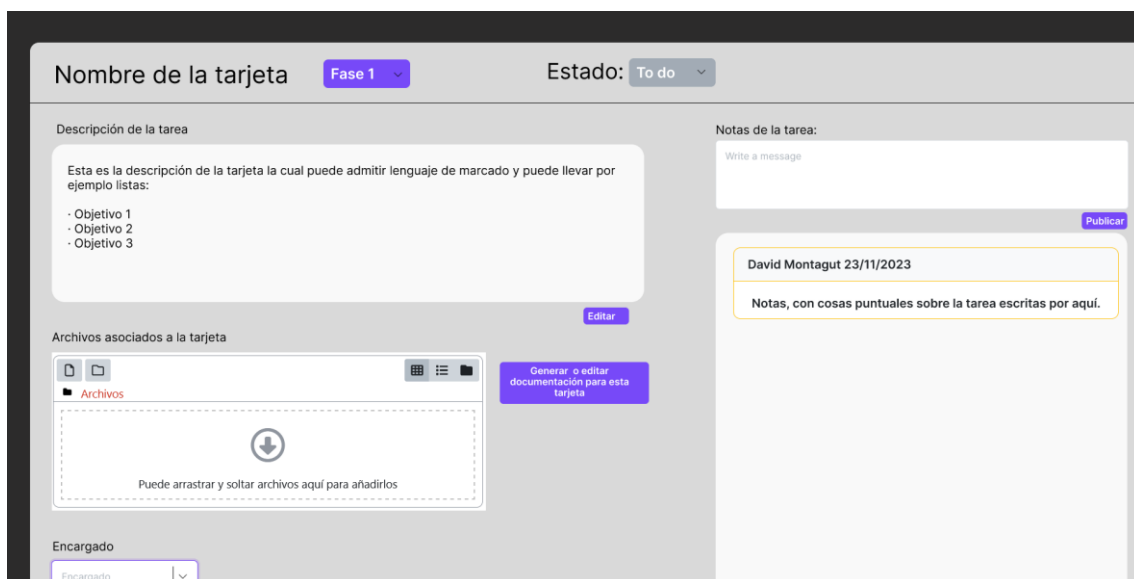


Figura 7 Ventana de una tarjeta

Esta ventana (figura 7) se planea que sea una nueva pestaña dentro del navegador. A través de esta ventana podremos acceder a toda la información de una tarea, una sección que describa la tarea, que archivos posee y los comentarios que se puedan ir añadiendo. Además se podrá crear la documentación asociada a esta tarjeta desde el botón de generar documentación mediante el editor de lenguajes de marcado de la figura 8.

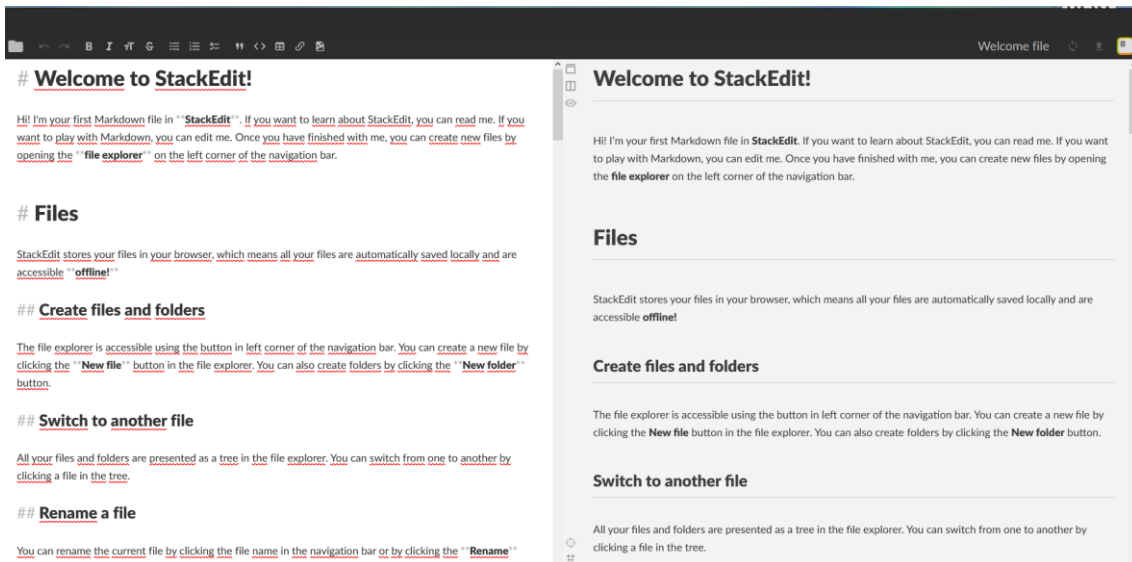


Figura 8 Editor de lenguajes de marcado

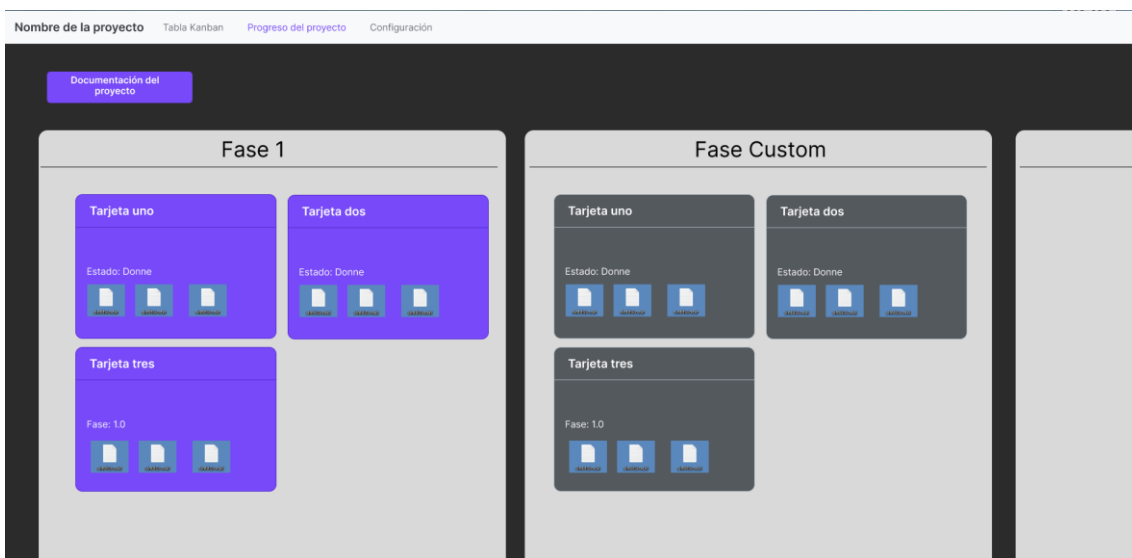


Figura 9 Menú del progreso del proyecto

La figura 9 muestra una vista que ayudará a ver el progreso de nuestro proyecto, por fases, insertando en cada fase que tareas se han realizado. Así podremos tener una visión más global del proyecto.

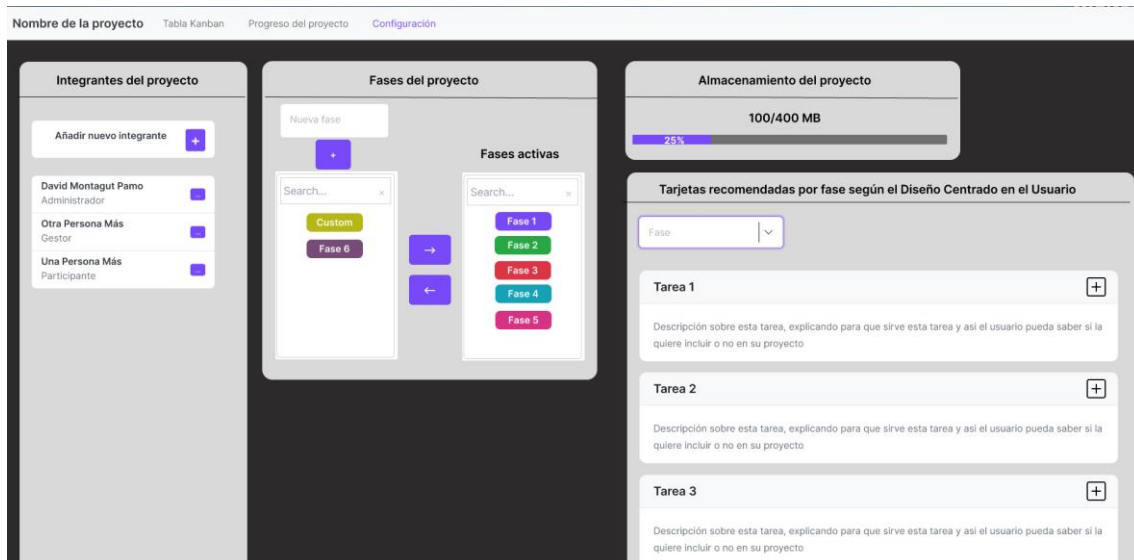


Figura 10 Configuración.

Por último, en la figura 10 accederemos a la configuración del proyecto donde podremos ver que usuarios forman parte, que fases tiene nuestro desarrollo y que tareas típicas del DCU se pueden realizar.

Una vez finalizado el prototipo, se llevaron a cabo una serie de entrevistas para validar su funcionalidad y usabilidad.

Entrevista:

Las preguntas y explicaciones en la entrevista fueron las siguientes:

- Te pongo en contexto de la aplicación: Se trata de una aplicación para gestionar proyectos que sigan la metodología del diseño centrado en el usuario. Que por si no lo sabes es una metodología para gestionar proyectos que se centra mucho en como interaccionan los usuarios con el sistema. Por lo tanto, todos los proyectos que se rigen por esta metodología se acaban dividiendo en varias fases como, por ejemplo, entrevistas con los usuarios, desarrollo de bocetos, pruebas con los usuarios ...
Entonces con esta aplicación se pretende abordar el desarrollo de un proyecto con estas características.
(Es necesario hacer una introducción a la aplicación porque, aunque los entrevistados son ingenieros pueden no tienen por qué saber sobre el DCU)
- Ahora mismo te encuentras en el menú principal del tu proyecto.
¿Te parece redundante que en el menú y en la barra superior te muestren las mismas opciones?
¿Entiendes lo que puedes encontrar en cualquiera de las 3 opciones?
(Lo primero es pensar en el menú principal del proyecto es el lobby desde donde se moverán los usuarios ha de ser cómodo para ellos y además han de entender hacia donde se mueven)

- Prueba a acceder a la tabla Kanban.
¿Has usado alguna vez este tipo de tablas en la gestión de un proyecto?
¿Qué información te estoy mostrando con esta vista?
(Es importante debido a la relevancia de la tabla Kanban incidir sobre el concepto y ver si es útil y comprensible)
- Intenta acceder a los detalles de la tarjeta 4.
Esta ventana yo la crearía en una ventana emergente nueva, tienes otra sugerencia Hay algo que no entiendas?
(Todo lo referente a la tabla Kanban ha de estar refinado y ha de ser cómodo para el usuario)
- Sal y accede a la configuración del proyecto.
Qué crees que se hace en el apartado de fases del proyecto.
(La configuración del proyecto no puede llevar a dudas ya que no queremos que las cosas se configuren mal)
- Por último, vuelve al menú principal.
Y accede a la ventana de progreso del proyecto.
Se te ocurre algo más que se puede mostrar en la preview de una tarjeta.
(Esta ventana fue una sugerencia realizada por alguien de confianza y hay que ver si es útil o no)
- Comentario personal.
(Conocer aquellas cosas que no hemos preguntado y los usuarios nos quieran comunicar)

Las entrevistas realizadas se pueden ver en el siguiente [anexo](#)

Conclusiones:

Las entrevistas realizadas con los usuarios fueron fundamentales para obtener retroalimentación valiosa sobre la usabilidad y funcionalidad de la aplicación desarrollada. Las respuestas obtenidas proporcionaron una visión clara de las áreas que funcionan bien y aquellas que requieren mejoras. A continuación, se presentan las principales conclusiones derivadas de estas entrevistas:

- En el primer menú se valora positivamente la existencia de redundancia con el navegador superior ya que así se mantiene el estilo por toda la página.
- El nombre de la opción “Tabla Kanban” puede no ser el mejor puesto que es un término muy exacto y la primera vez que te lo encuentres puedes no saber qué es.
- Para acceder a los datos de una tarjeta es mejor poder hacer clic sobre toda la tarjeta, no solo sobre un botón.
- Es preferible que los datos de las tarjetas sean visibles desde la misma ventana que se invocan usando una ventana modal.
- Para cada tarjeta, poder implementar un registro de los cambios realizados en la misma.

- El apartado de “notas” no se entiende del todo por lo tanto sería mejor distribuirlo de otra manera.
- En la ventana de configuración añadir más campos como: cambio del nombre del proyecto o poder crear otras columnas de la tabla Kanban.
- La ventana de configuración podría estar mejor organizada dejando los datos más irrelevantes en la parte inferior de la pantalla.
- El termino de fases del proyecto no se entiende del todo, buscar una palabra que lo defina mejor, o ser capaces de explicarlo.
- Posible implementación de un tutorial que indique las funcionalidades de la aplicación ya que muchos usuarios no saben para que sirven los botones referentes a la generación de documentación.
- Para la ventana de progreso del proyecto es mejor cambiar el icono por uno que se asemeje más al formato mostrado.
- También en esa misma ventana en la preview de las tarjetas, estaría bien mostrar un resumen de la descripción de la tarjeta, además de destacar con algo más visual en qué estado se encuentra cada tarea.
- Esa misma ventana tampoco se ve muy útil y es una posibilidad su eliminación del proyecto.

Desarrollo de la aplicación:

Este es el apartado que más tiempo de desarrollo necesitó abarcando prácticamente desde enero hasta mayo de 2024.

Todo comienza con un primer diseño de la base de datos:

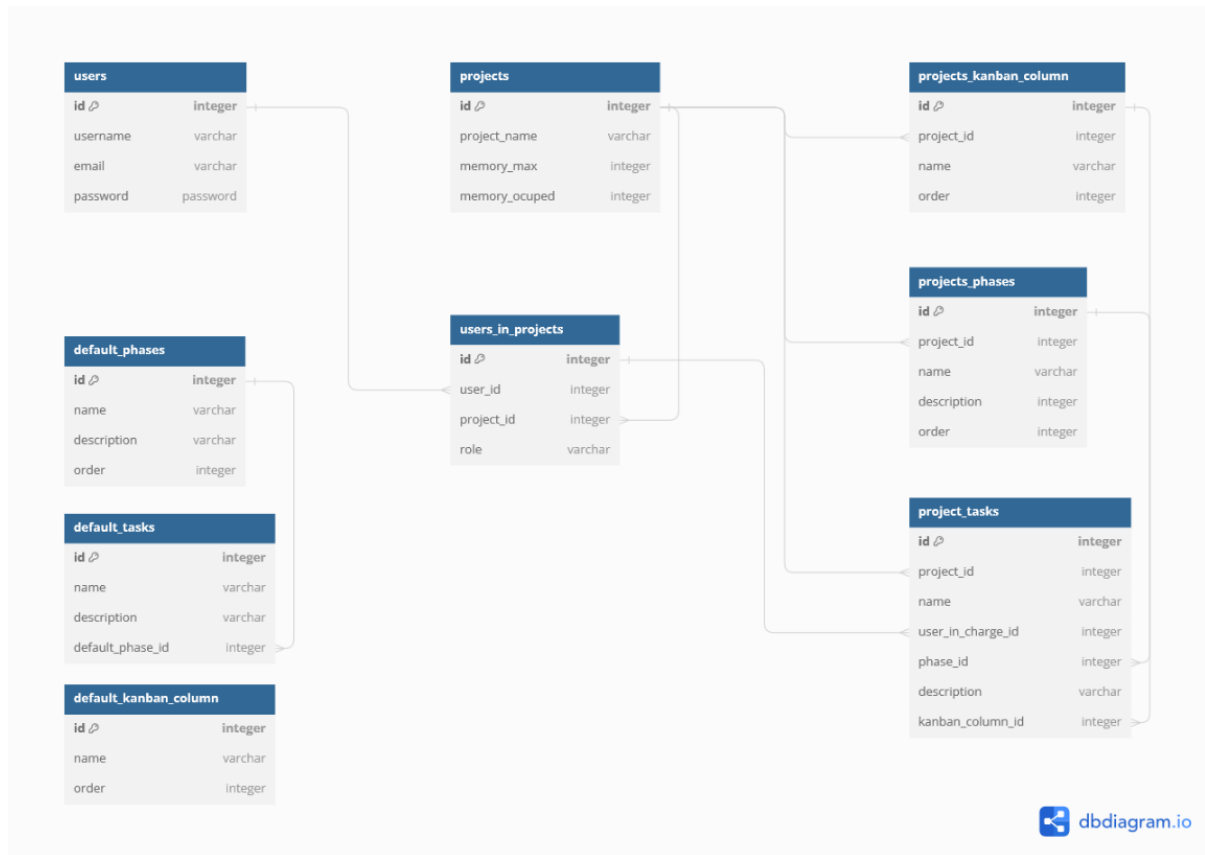


Figura 11 Base de datos inicial

Explicación de las distintas tablas (Ver figura 11):

- **Users**: Donde se almacenarán los usuarios de la web
- **Projects**: Identificará los proyectos que podrán iniciar los usuarios
- **Users in Project**: Tabla que vincula los usuarios con los proyectos de los que forman parte
- **Project_kanban_columns**: Aquellas columnas de la tabla Kanban creadas por los usuarios
- **Project_phases**: Aquellas fases en las que se podrá dividir el proyecto y se verán reflejadas en la tabla Kanban.
- **Project_task**: Las tareas en las que se dividirá el proyecto.

Tablas default: Contendrán aquellas, fases, tareas y columnas que tendrá la aplicación, rellenadas por defecto con información estática.

La etapa de codificación dio como resultado la siguiente aplicación:

Resultado en [video de la primera versión de la aplicación](#)

Aquí resaltare algunos de los aspectos más destacables:

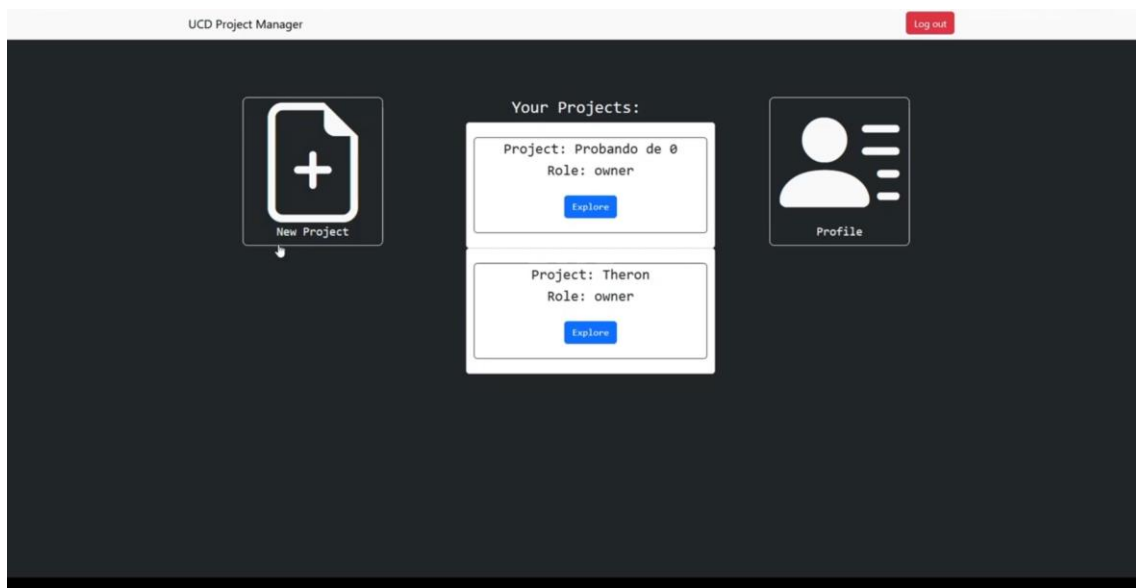


Figura 12 Menú principal

Tras el Login este es el menú (figura 12) desde dónde podremos visualizar nuestros proyectos y nuestro perfil.



Figura 13 Menú del proyecto

Una vez seleccionemos un proyecto veremos un menú (figura 13) desde donde puede acceder a la tabla Kanban, los documentos del proyecto y la configuración del proyecto.

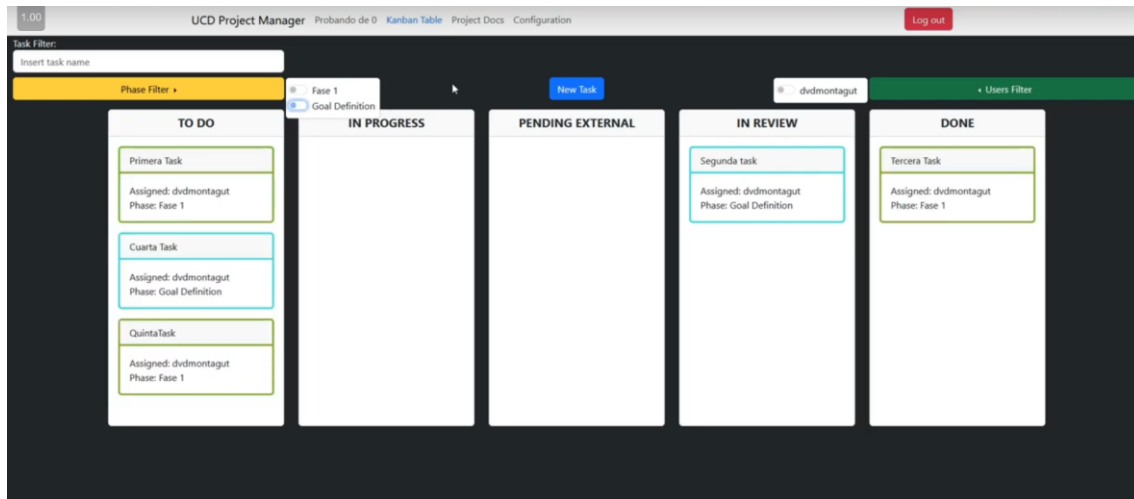


Figura 14 Tabla Kanban

Este es el resultado de la tabla Kanban (figura 14). La posición de los filtros y como marcar los seleccionados además del color de las tarjetas cambió con respecto a el prototipo en Figma debido a la inexperiencia con las hojas de estilo del programador.

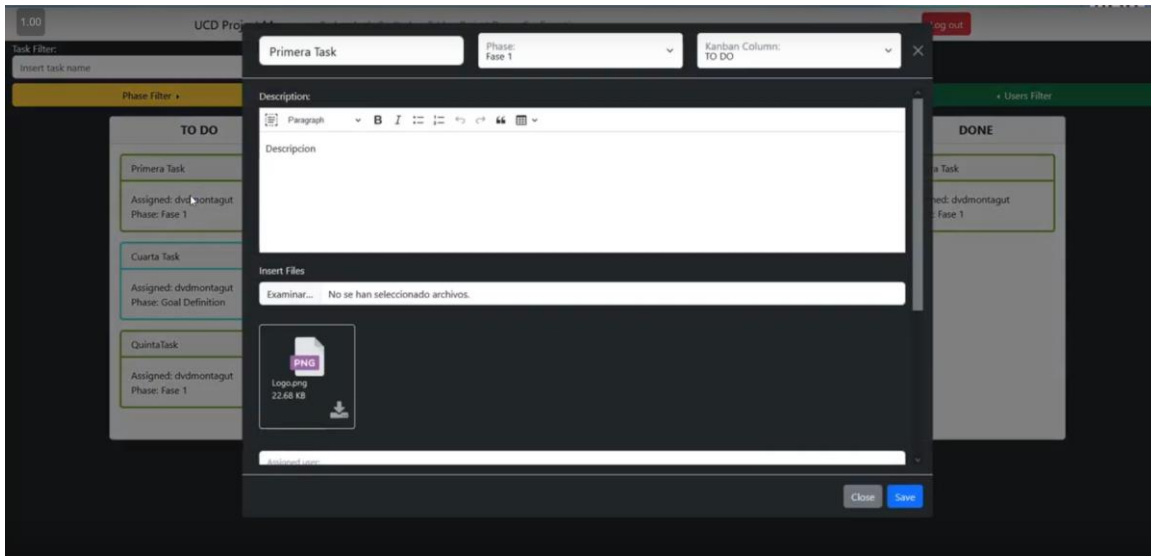


Figura 15 Ver una tarjeta Parte1

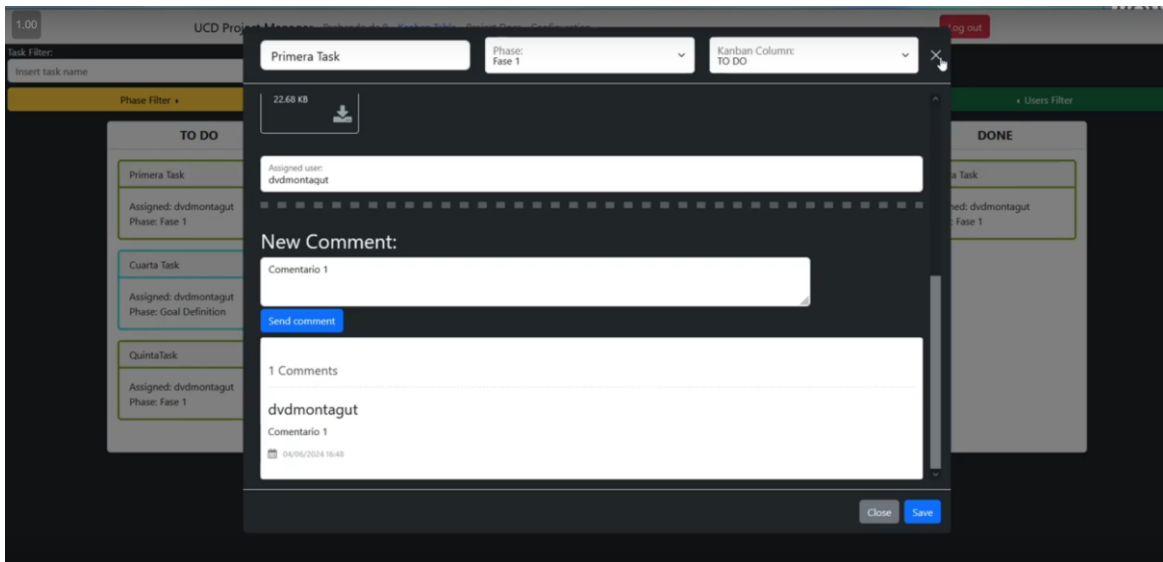


Figura 16 Ver una tarjeta Parte2

Otro apartado que sufrió cambios fue la forma de verse una tarjeta (figuras 15 y 16) por dentro debido a que el tamaño con el que se diseñaron los apartados internos a la hora de mostrarlos en una ventana modal que no ocupase toda la pantalla se hacían muy ilegibles.

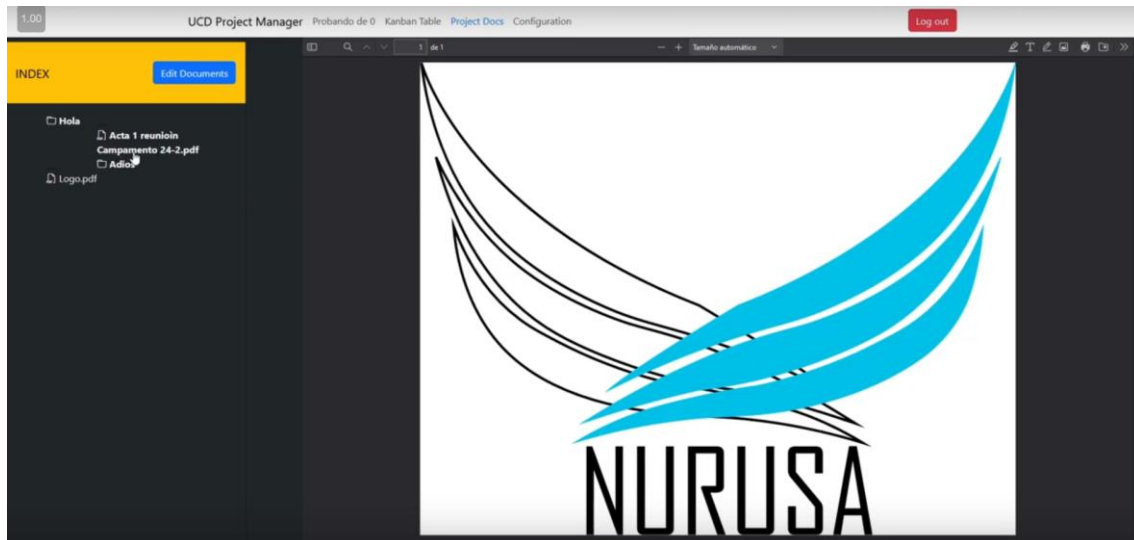


Figura 17 Apartado para la documentación

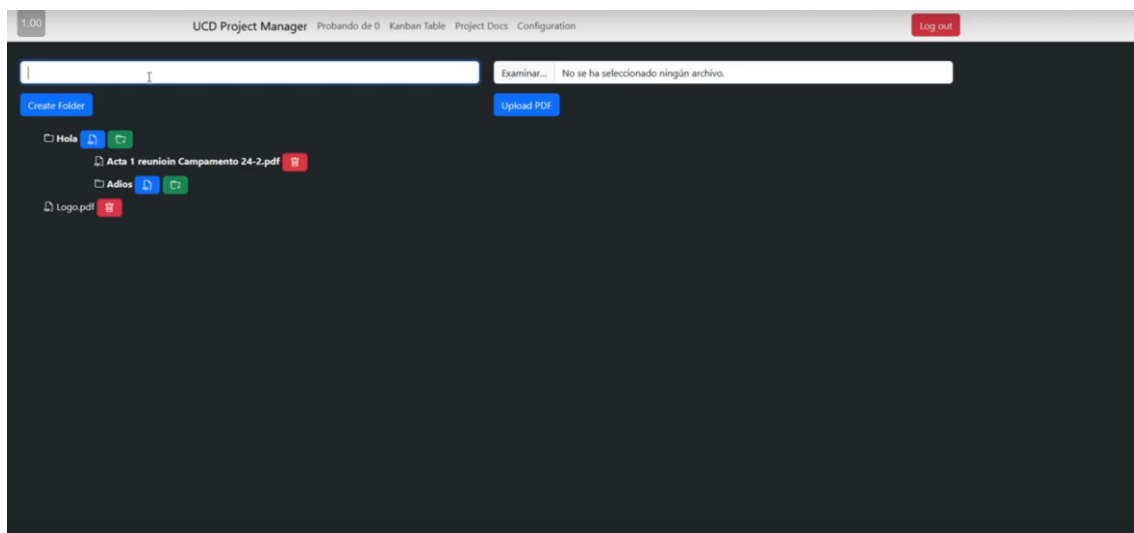


Figura 18 Apartado para organizar los archivos

Por motivos de tiempo el apartado de la documentación usando lenguajes de marcado y creando una wiki se tuvo que variar a un diseño más simple (figuras 17 y 18). En el que se permitiría subir archivos PDF de forma estructurada en forma de ficheros. Este fue el apartado que más varió tanto a nivel estético como a nivel de funcionamiento del prototipo a la implementación.

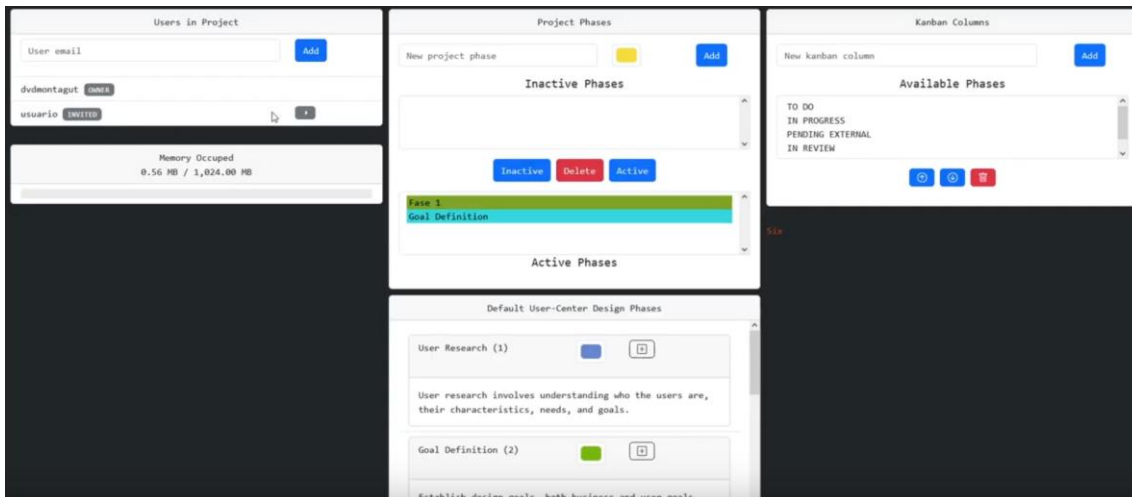


Figura 19 Configuración del proyecto

La configuración del proyecto es el apartado que menos cambios recibió en comparación con el prototipo en Figma.

Esta fue la primera versión funcional del proyecto. Con esto creado se mantuvo una reunión con el tutor del TFM. De esa reunión se extrajeron los siguientes puntos.

- En el menú inicial mostrar que hay que acceder a la configuración del proyecto para poder usar la tabla Kanban. (Esto es debido a que para poder crear “Project phases” primero hay que declararlas en la configuración)
- A la hora de añadir usuarios a tu proyecto, poder invitar a usuarios que no estén previamente registrados en el proyecto. (Esto no se llegó a implementar pero se planteará como trabajo futuro)
- Modificación del apartado de fases activas para que siga la misma línea de diseño donde los epígrafes estén arriba.

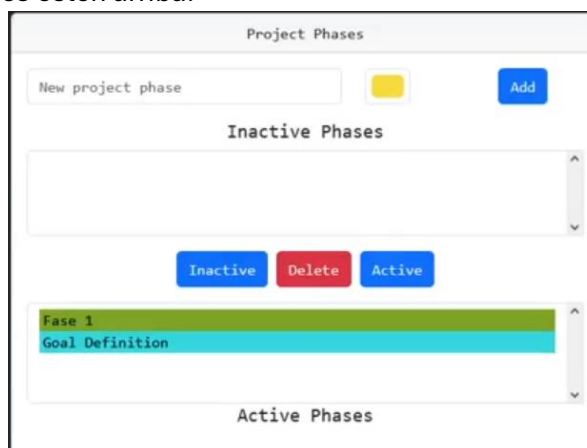


Figura 20 Hay que hacer que Active Phases este debajo de los botones

- Reestructuración de la funcionalidad del proyecto. Necesidad de implementar un sistema para poder ver en que iteración se encuentra el proyecto y todo lo que se ha ido realizando en él sustentando en las fases del diseño centrado en el usuario.
- Eliminación del apartado para subir documentos.
- Posible cambio del público objetivo a estudiantes del grado de informática.

Fase de *Discovery* 2:

Tras la reunión previa con el tutor y hacer una breve planificación del tiempo restante hasta la fecha de entrega. Se opta por cambiar el enfoque del público objetivo que nos había recomendado la IA.

El público ya no serían desarrolladores y gerentes de proyectos. Si no que se enfocaría hacia usuarios novatos con el uso del Diseño Centrado en el Usuario como pueden ser los alumnos de la Universidad que cursen asignaturas relacionadas con ello.

Perfiles de Usuario

Perfil 1: Estudiante de Grado en Informática

- **Nombre:** Laura Gómez
- **Edad:** 21 años
- **Género:** Femenino
- **Ubicación Geográfica:** Salamanca, España
- **Nivel de Educación:** Estudiante de tercer año en Ingeniería Informática
- **Ocupación:** Estudiante universitaria
- **Características Personales:** Creativa, curiosa, con habilidades técnicas básicas, busca aprender y aplicar nuevas tecnologías.
- **Necesidades:** Acceso a recursos educativos, herramientas colaborativas, guía en la aplicación de metodologías DCU.
- **Objetivos:** Completar proyectos académicos con éxito, adquirir experiencia práctica en DCU, mejorar sus habilidades técnicas.
- **Desafíos:** Falta de experiencia práctica, necesidad de equilibrar el estudio con otros compromisos, comprensión de conceptos avanzados de DCU.

Segmentar los Usuarios

Segmento 1: Estudiantes de Informática y Diseño

Características:

- Edades entre 18 y 25 años.
- Estudiantes de grados relacionados con la informática y el diseño.

Necesidades Comunes:

- Acceso a recursos educativos y guías prácticas sobre DCU.
- Herramientas colaborativas para trabajar en equipo.
- Recursos para la creación y prueba de prototipos.

Objetivos Comunes:

- Aplicar metodologías DCU en proyectos académicos.
- Adquirir experiencia práctica y mejorar habilidades técnicas y de diseño.
- Completar proyectos con éxito y desarrollar un portafolio sólido.

Fase de Prototyping and Testing 2:

En este punto se transformó la aplicación de forma que se hicieron presentes los siguientes objetivos:

- Separación de la tabla Kanban del sistema de fases y subfases del DCU.
- Implementación de un sistema para poder crear las fases y subfases del DCU.
- Poder visualizar las iteraciones de tu propio proyecto.
- Eliminación del sistema de ficheros.

La aplicación resultante se puede ver [aquí](#)



Figura 21 Menú del proyecto

Ahora el menú (ver figura 21) consta de un nuevo botón que te lleva al menú del ciclo de proyecto.



Figura 22 Menú del ciclo del proyecto

Este es el menú (ver figura 22) del ciclo del proyecto desde el cual se podrá acceder a todas las opciones referentes a las iteraciones, fases, subfases o metodologías que se pueden seguir en un proyecto basado en DCU.

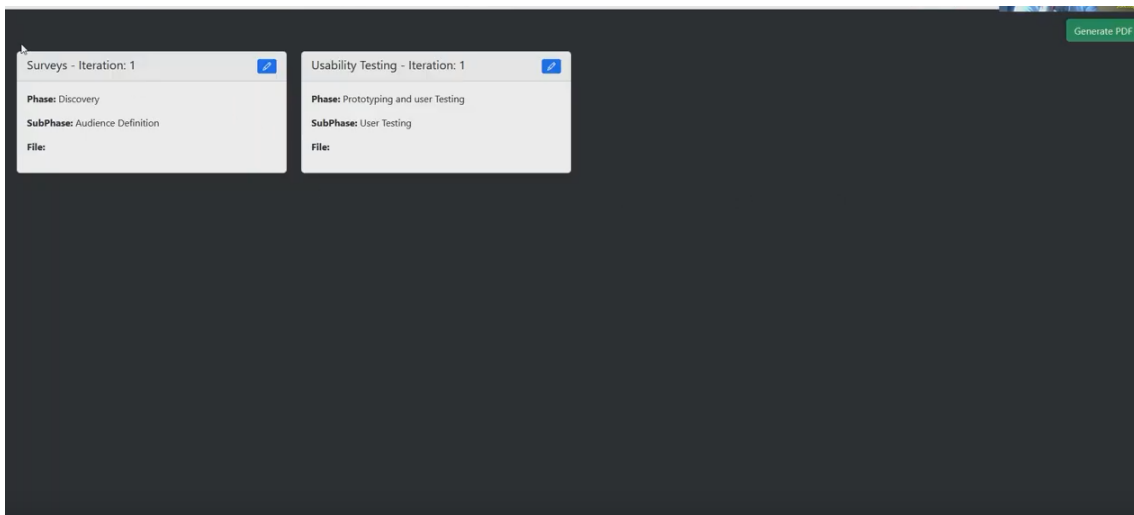


Figura 23 Pantalla del ciclo del proyecto

En esta ventana (ver figura 23) podremos ver qué pasos hemos ido realizando dentro de nuestro proyecto en orden. Además, podremos responder a varias preguntas para así poder generar la documentación asociada al proyecto

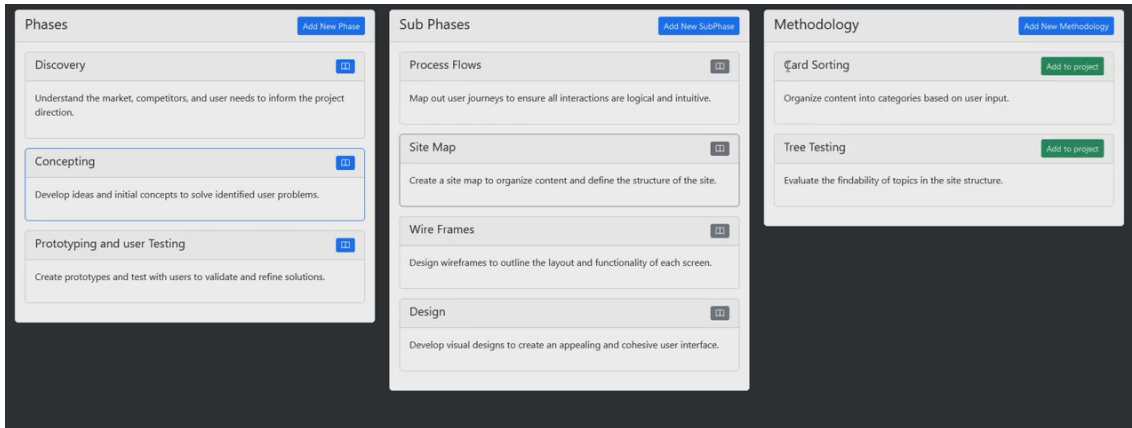


Figura 24 Pantalla de selección de las fases

En esta pantalla (ver figura 24) se puede ver que opciones tenemos a la hora de afrontar el ciclo de nuestro proyecto. Todas cuentan con una concisa descripción. Además de que podremos crear nuevas fases, subfases o metodologías a nuestra preferencia.

Entrevista de concepto

Después de realizar los primeros bocetos y obtener la aprobación del tutor, se llevó a cabo una prueba con un usuario que había interactuado previamente con el prototipo de Figma. Durante la prueba, el usuario utilizó la aplicación mientras se le hacían preguntas para evaluar su experiencia. Desafortunadamente, la grabación de esta entrevista solo capturó los primeros minutos debido a un problema técnico.

Conclusiones obtenidas:

- Botón de Registro en la Ventana de Login: Añadir un botón en la ventana de login que redirija a la página de registro.
- Mensaje de Clasificación de Tarjetas Kanban: Incluir un mensaje tanto en el menú como en la pantalla de la tabla Kanban que indique la necesidad de crear una clasificación para las tarjetas Kanban antes de poder crearlas.
- Pop-up de Error: Utilizar un pop-up para informar al usuario cuando una acción no se ha realizado correctamente.
- Terminología "Tag": Usar el término "Tag" para la clasificación de las tarjetas Kanban, lo cual fue considerado una mejor opción.
- Múltiples Tags por Tarjeta: Permitir que una tarjeta pueda tener varios tags.
- Movimiento de Tarjetas: Le pareció correcta la forma de mover y desplazar las tarjetas.
- Eliminación de Tarjetas: Las tarjetas deben ser eliminables, con un botón para borrar dentro de la propia tarjeta.
- Orden de Comentarios: Mostrar los comentarios en orden cronológico, del más antiguo al más reciente.
- Filtros de Tarjetas: Colocar todos los filtros en la misma área, ya sea a la izquierda o en la parte superior, para mejorar la claridad y usabilidad.
- Iconos del Ciclo del Proyecto: Revisar los iconos utilizados, ya que algunos no transmitían correctamente su función, uno de ellos parece el diagrama de Gant.
- Unificación de Dashboards: Unificar los dos dashboards en una sola ventana, según la propuesta del entrevistado.

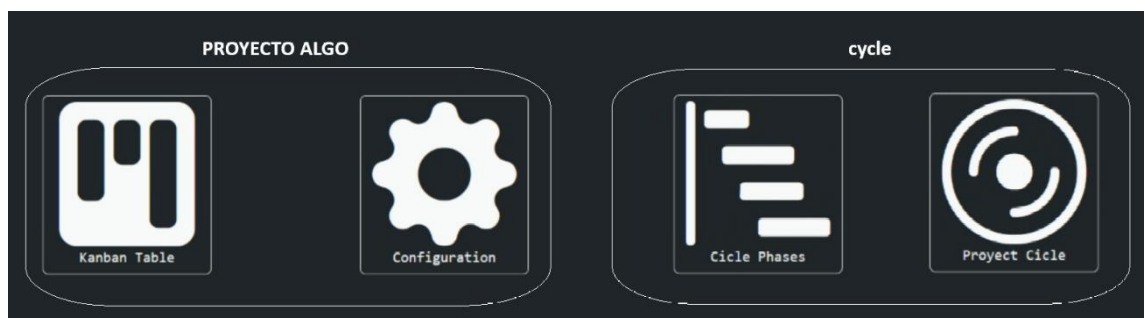


Figura 25 Propuesta del entrevistado

- Botones en la Ventana de Fases y Subfases: Asegurar que los botones sean siempre del mismo color en fases y subfases, y hacer más claro el modal para añadir metodologías al proyecto.
- Explicación de Términos: Incluir tooltips para términos como "iteración".
- Errores de Programación: Solucionar errores que impedían respuestas largas en las preguntas anexas a la metodología.
- Visibilidad del Botón de Generar PDF: Hacer más visible el botón de generar PDF.
- Filas para Iteraciones: Crear filas para cada iteración para visualizar el proyecto de forma más clara.

Tras aplicar estas sugerencias, la aplicación fue actualizada y los cambios se pueden ver en el [siguiente video](#)

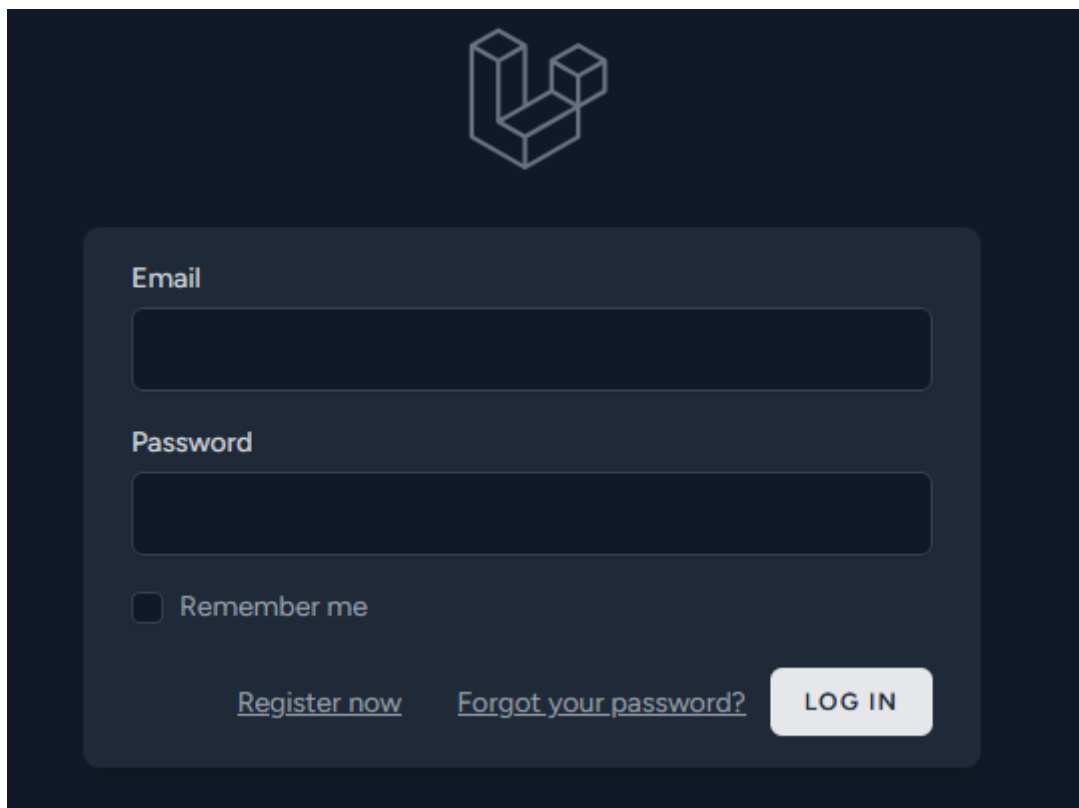


Figura 26 Ventana de log in

El login (figura 26) ahora cuenta con una opción para registrarse.

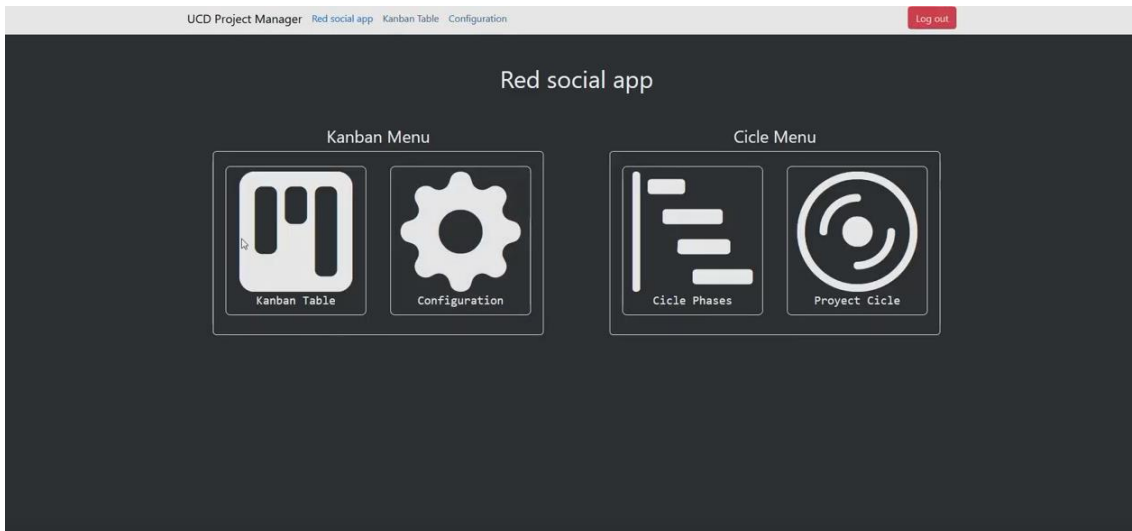


Figura 27 Menú del proyecto

Este es el menú del proyecto una vez aplicados los cambios (figura 27) para que ahora el menú solo sea uno dividido en dos apartados. Y además ahora sale una notificación para obligarte a pasar por el apartado de configuración antes de ir a la tabla Kanban.

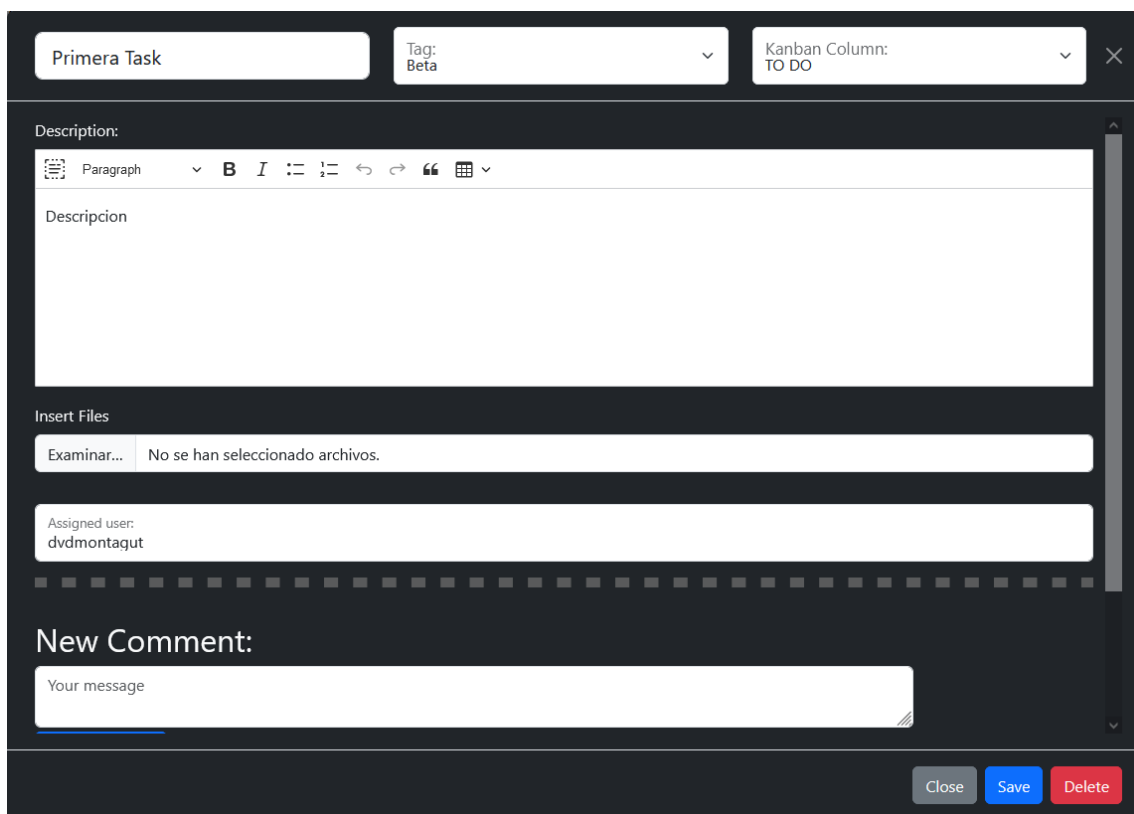


Figura 28 Información de una tarjeta

Ahora las tarjetas (figura 28) pueden ser eliminadas y se ha implementado el termino de *Tag* para clasificar las tarjetas.

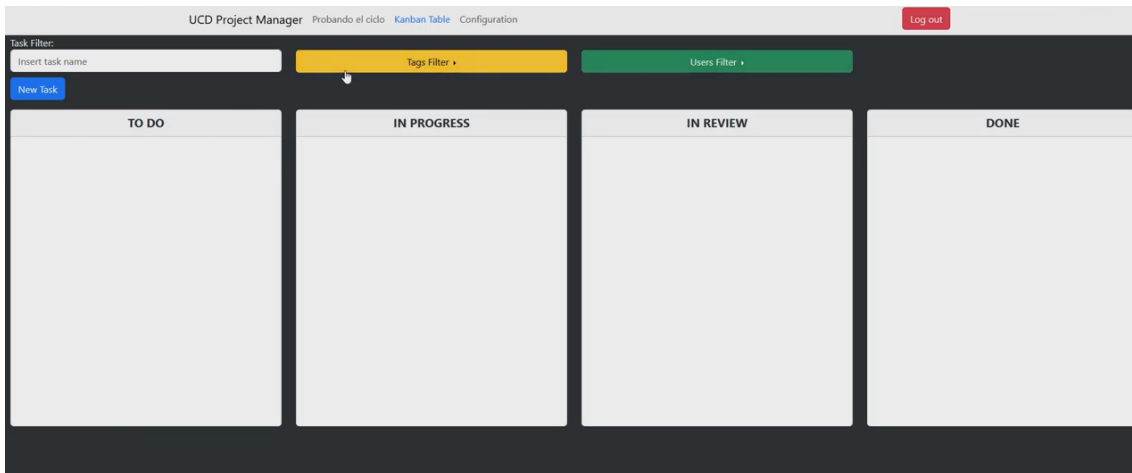


Figura 29 Tabla Kanban

Se han recolocado los filtros (figura 29) para que ahora todos estén en la misma altura otorgando así uniformidad a su significado.

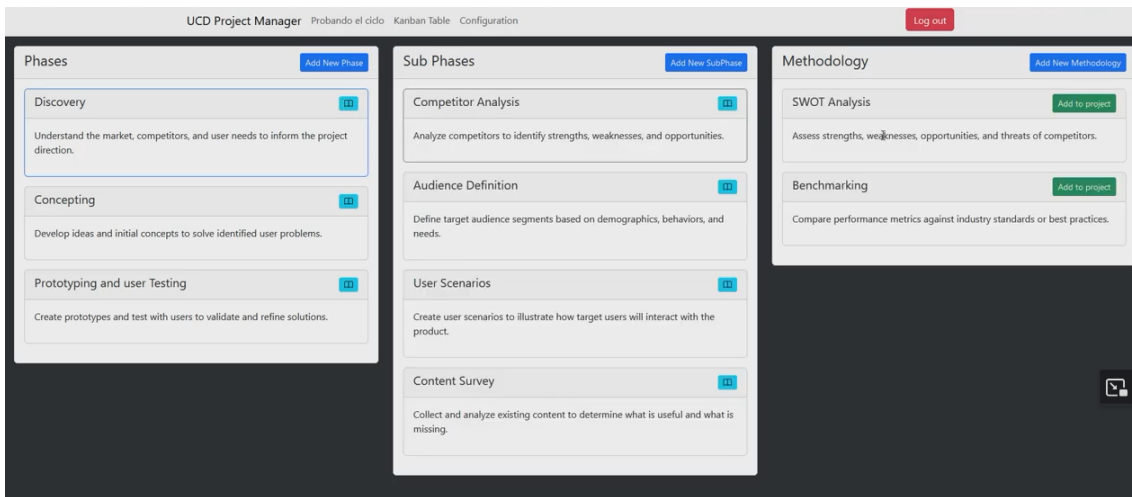


Figura 30 Menú para ver las fases, subfases y metodologías

Ahora los botones para desplegar las subfases y las metodologías (figura 30) se ven todos con el mismo color.

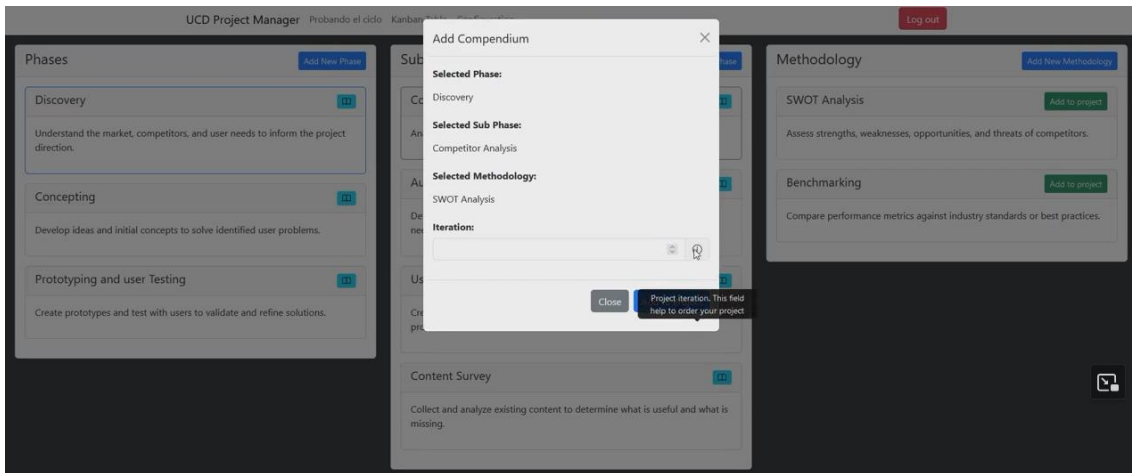


Figura 31 Modal para poder añadir una metodología a las iteraciones

A la hora de añadir una nueva metodología se han hecho más legibles los datos (figura 31) que se han seleccionado y, además. Y se ha añadido un mensaje para hacer más entendible el termino de iteración.

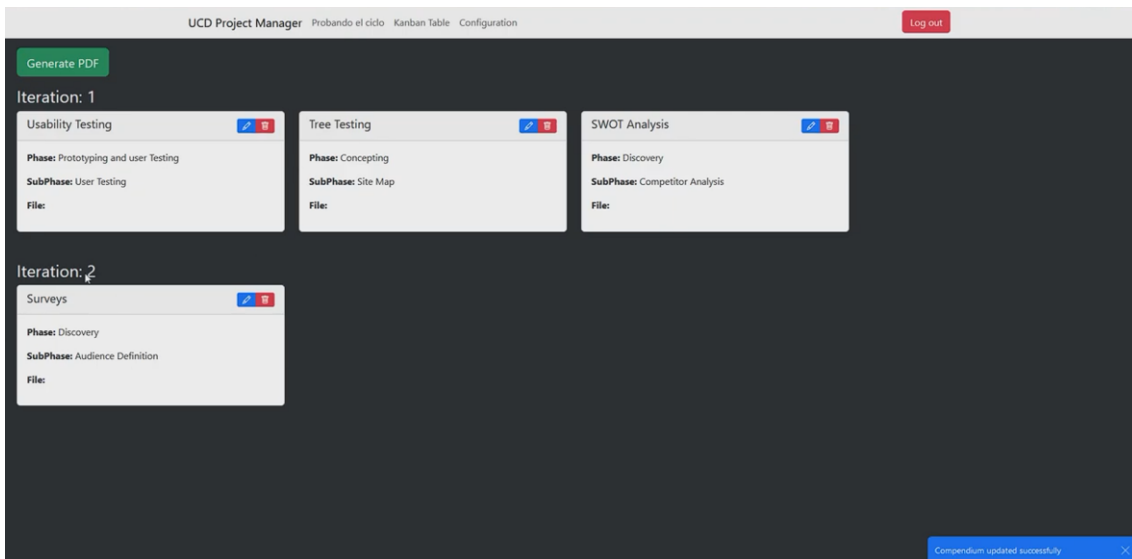


Figura 32 Ventana para ver el ciclo de tu proyecto

La ventana para ver las iteraciones ahora muestra el botón para generar la documentación a la izquierda. Cada iteración tiene su propia fila. Y como se puede apreciar aparecen *Pop Ups* en la parte inferior derecha cuando una acción se realiza.

Entrevista para comprobar los iconos usados en la aplicación.

Tomando como punto de partida la anotación de la última entrevista: “Los iconos del apartado del ciclo del proyecto no transmiten lo que deben, uno le ha parecido un diagrama de Gantt y el otro un CD”, se decidió realizar dos entrevistas adicionales con otros usuarios para evaluar la efectividad de los iconos empleados en la aplicación. Durante estas entrevistas, se mostró a los usuarios cada icono utilizado en la aplicación y se les preguntó si podían entender su propósito sin ninguna explicación previa.

El concepto de *affordance*, introducido por el psicólogo James J. Gibson y popularizado en el diseño de interfaces por Donald Norman, se refiere a las propiedades de un objeto que sugieren cómo debe ser utilizado. En el contexto de la interfaz de usuario, un buen *affordance* implica que el propósito de un icono o un elemento de la interfaz sea inmediatamente claro y comprensible para el usuario sin necesidad de instrucciones adicionales. Los resultados de las entrevistas iniciales indicaron que los iconos no estaban cumpliendo con este principio, ya que no comunicaban eficazmente su función a los usuarios.

Durante las entrevistas adicionales, se evaluó si los usuarios podían identificar correctamente la función de cada icono basado en su diseño y contexto dentro de la aplicación. Este enfoque ayudó a determinar qué iconos necesitaban ser rediseñados para mejorar su *affordance*, asegurando que los usuarios puedan interactuar con la aplicación de manera intuitiva y eficiente.

Las entrevistas [son estas](#)

De aquí se pudieron obtener varias conclusiones:



Figura 33 Icono antiguo para el ciclo del proyecto

Este botón (figura 33) no muestra de forma literal la forma en la que se verá la siguiente ventana, aunque si haga referencia a un ciclo (además está mal escrito).

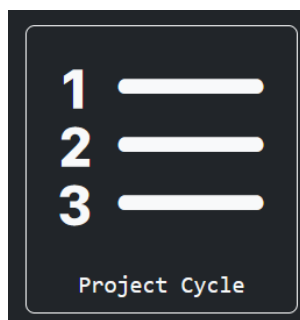


Figura 34 Nuevo icono para el ciclo del proyecto

Se cambia por este otro icono (figura 34), que hace referencia a las iteraciones que se van a poder ver dentro del ciclo del proyecto.

Otro cambio que se propone es el del icono parecido a un libro.

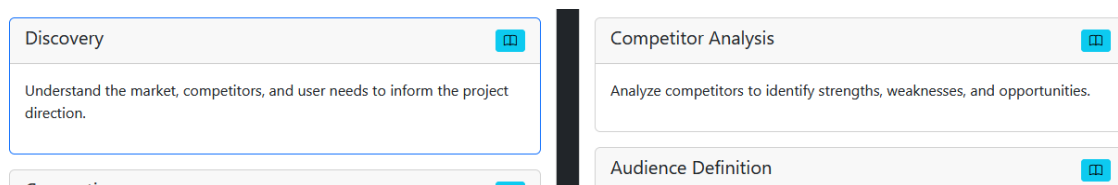


Figura 35 Icono para desplegar las subfases o las metodologías

El icono al ser pequeño puede no parecer pinchable y además puede transmitir que se va a mostrar una descripción más detallada sobre el punto en cuestión (figura 35).

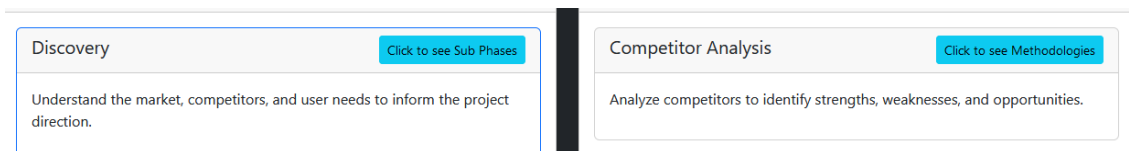


Figura 36 Cambio de iconos a botones con texto

Con la implementación de estos cambios, el proceso iterativo sobre el diseño del proyecto se considera concluido. La aplicación está lista para su entrega final, y de ahora en adelante, solo se realizarán optimizaciones en el código y correcciones de errores.

Cómo probar la aplicación.

Enlace a la aplicación:

<http://104.248.102.93:8000/>

- Usuario de prueba con el propio proyecto del TFM:
 - Usuario: tfmuser@yopmail.com
 - Contraseña: tfmuser123.
- Otros usuarios:
 - Usuario:
 - tfmuser1@yopmail.com
 - tfmuser2@yopmail.com
 - Contraseña: tfmuser123

Evaluación heurística

En este apartado se mando a varios usuarios probar la aplicación por libre fijándose en una serie de aspectos aspectos y puntuándolos con una nota del uno al cuatro además de poder añadir algún comentario.

A continuación, se explicarán los distintos aspectos y seguidamente se pondré la media de las puntuaciones obtenidas y los comentarios al respecto.

- 1. Visibilidad del estado del sistema:** El sistema debe mantener informados a los usuarios sobre lo que está ocurriendo, proporcionando retroalimentación adecuada dentro de un tiempo razonable. Por ejemplo, mostrar un indicador de carga cuando una página está procesando.
 - **Nota:** 2
 - **Comentarios:** “No se avisa del uso de las cookies.”
“Al intentar crear diferentes ítems, si no se cumple alguna condición el sistema no hace nada y no da mensajes de error, no pasa con todo.”
“Algunas acciones complejas desembocan en acciones no permitidas y no se explica al usuario el por qué detrás, Por ejemplo, borrar tags es bloqueado cuando existe alguna tarea con ese tag. ”
- 2. Alineación entre el sistema y el mundo real:** El sistema debe hablar el lenguaje de los usuarios, con palabras, frases y conceptos familiares para ellos, en lugar de términos orientados al sistema. Debe seguir convenciones del mundo real y presentar la información en un orden natural y lógico.
 - **Nota:** 4
 - **Comentarios:** No hay.
- 3. Control y libertad para el usuario:** Los usuarios a menudo eligen funciones del sistema por error y necesitarán una "salida de emergencia" claramente marcada para salir del estado no deseado sin tener que pasar por un diálogo extendido. Esto se traduce en ofrecer deshacer y rehacer.
 - **Nota:** 3
 - **Comentarios:** “Algunas opciones no se pueden deshacer. Por ejemplo, borrar un proyecto.”
“Borrar Tags es laborioso si hay muchas tareas asignadas a ese tag.”
- 4. Consistencia y estándares:** Los usuarios no deberían tener que preguntarse si diferentes palabras, situaciones o acciones significan lo mismo. La consistencia debe mantenerse a través de la interfaz, y el uso de estándares debe ser claro para los usuarios.
 - **Nota:** 4
 - **Comentarios:** “La barra de menú es práctica, pero a su vez le falta un botón para acceder al perfil. ”

5. **Prevención de errores:** En lugar de tener buenos mensajes de error, un diseño cuidadoso debe prevenir que ocurran problemas en primer lugar. Se deben eliminar condiciones propensas a errores o verificarlas y ofrecer a los usuarios una opción de confirmación antes de que actúen.
 - **Nota:** 3
 - **Comentarios:** “La prevención de error es correcta y te previene de hacer muchas acciones si pudiesen introducir errores en el sistema. Sin embargo, no hay diálogos de doble confirmación en algunas opciones de borrado.”
“Existe el requisito de tener tags activos para acceder al Kanban table. Sin embargo, se puede acceder desde la barra superior. También si tenemos tags inactivos.”

6. **Reconocimiento antes que recuerdo:** Minimizar la carga de memoria del usuario haciendo visibles los objetos, acciones y opciones. El usuario no debería tener que recordar la información de una parte del diálogo a otra. Las instrucciones para el uso del sistema deben ser visibles o estar fácilmente disponibles cuando sea necesario.
 - **Nota:** 4
 - **Comentarios:** No hay.

7. **Flexibilidad y eficiencia de uso:** Los aceleradores (accesos directos) pueden ayudar a los usuarios expertos a moverse rápidamente. El sistema debe poder adaptarse a usuarios principiantes y avanzados, permitiendo que cada grupo utilice la aplicación a su manera.
 - **Nota:** 3
 - **Comentarios:** “Pese a que tenemos la barra superior que nos permite acceder a todos lados sería interesante poder usar atajos de teclado para eliminar/copiar tareas, por ejemplo. ”

8. **Estética y diseño minimalista:** Las interfaces no deben contener información irrelevante o raramente necesaria. Cada unidad extra de información compite con las unidades relevantes de información y disminuye su visibilidad relativa.
 - **Nota:** 4
 - **Comentarios:** No hay.

9. **Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores:**

Los mensajes de error deben expresarse en un lenguaje claro, indicando precisamente el problema y sugiriendo una solución constructiva.

- **Nota: 1**
- **Comentarios:** “Los mensajes de error no describen el problema. Dicho problema puede no estar relacionado con lo que tenemos en pantalla si no otras dependencias, por ejemplo, no poder borrar un tag porque una tarea lo posee.”
“En estos casos es necesaria la información adicional ya que tras varias iteraciones podemos no saber por qué ni cómo solucionarlos.”

10. **Ayuda y documentación:** Aunque es mejor si el sistema se puede utilizar sin documentación, puede ser necesario proporcionar ayuda y documentación. Esta información debe ser fácil de buscar, centrada en la tarea del usuario, listar pasos concretos a seguir y no ser demasiado extensa.

- **Nota: 3**
- **Comentarios:** “Un usuario informado acerca de proyectos centrados en el usuario entenderá perfectamente la aplicación.”
“Sería interesante en la página de creación de proyectos detallar los pasos o tener una página de introducción que resuma brevemente los apartados disponibles y sus posibilidades.”

Conclusiones y trabajo futuro.


La realización de este trabajo ha permitido obtener varias conclusiones significativas:

1. **Beneficios del Proceso Iterativo del DCU:** El seguimiento de un proceso iterativo como el Diseño Centrado en el Usuario (DCU) ha sido fundamental para refinar la herramienta de maneras que no habrían sido posibles con un proceso de diseño secuencial. Este enfoque iterativo ha permitido una evaluación continua y ajustes basados en la retroalimentación de los usuarios, resultando en una aplicación más alineada con las necesidades reales de los usuarios como es el ejemplo de la autogeneración de documentación.
2. **Optimización de Funcionalidades:** A través del proceso del DCU, se pudieron identificar y eliminar funcionalidades que no aportaban valor real al proyecto. Esto no solo simplificó la aplicación, sino que también permitió rediseñar y enfocar mejor las funcionalidades existentes, como la tabla Kanban, mejorando su utilidad y eficiencia.
3. **Aumento de la Seguridad y Eficacia del Proyecto:** La aplicación del DCU incrementó la confianza en el proyecto final, ya que cada decisión de diseño fue probada y validada por usuarios reales. Este enfoque aseguró que la herramienta es intuitiva y comprensible, minimizando la necesidad de explicaciones adicionales sobre las decisiones de diseño, ya que estas fueron basadas en datos y retroalimentación de usuarios. Por ejemplo la selección de iconos al haber pasado por varias pruebas debería ser explicativa para la mayoría de usuarios.
4. **El uso de la inteligencia artificial como complemento del DCU:** La inteligencia artificial ha sido de gran ayuda para llegar a aproximaciones con realidades en las que no estaba tan familiarizado como por ejemplo la definición de usuarios. Aun así, al recurrir tanto a tópicos es muy poco exacta en cuanto más te acercas a un problema concreto ya que a no ser que lo expliques todo de manera exacta, no es capaz de saber todo el contexto que necesitas. Aun así, te puede ayudar a enfocarte de manera muy gratificante.

Trabajo Futuro

Aunque la herramienta ha sido probada con usuarios durante las fases de prueba, la implementación en un escenario real puede revelar nuevos desafíos y áreas de mejora. Los siguientes puntos destacan áreas potenciales para el trabajo futuro:

1. **Monitoreo en Entornos Reales:** Es esencial seguir monitoreando la herramienta en un entorno real para identificar y corregir cualquier mal funcionamiento que pueda surgir. La observación continua permitirá realizar ajustes necesarios para garantizar una operación óptima y una experiencia de usuario fluida.
2. **Mejoras en el Diseño Visual:** La cohesión visual es crucial para la usabilidad y la estética de la herramienta. Se recomienda revisar y posiblemente modificar elementos visuales para asegurar una mayor coherencia en el estilo de la interfaz. Esto puede incluir ajustes en la paleta de colores, tipografía y disposición de elementos para mejorar la experiencia visual y de uso.
3. **Extensión de Funcionalidades:** Con base en la retroalimentación de los usuarios en escenarios reales, se pueden identificar nuevas funcionalidades que podrían ser añadidas para aumentar el valor de la herramienta. La adopción de un enfoque iterativo continuará siendo valiosa para integrar estas nuevas características de manera efectiva.
4. **Mejoras en la seguridad del sistema:** Durante el tiempo de pruebas la base de datos de la aplicación sufrió un ataque. En el ataque accedieron de forma completa a ella. De esta manera:
 - a. Acceso al servidor por SSH sólo por certificado digital en vez de contraseña.
 - b. Luego capa con IP Tables el puerto de escucha del MariaDB para que sólo puedas acceder en localhost.



```
MariaDB [aplicacion]> SELECT * FROM README;
+-----+-----+-----+-----+
| id | Message | Bitcoin_Address |
+-----+-----+-----+-----+
| 1 | I have backed up all your databases. To recover them you must pay 0.007 BTC (Bitcoin) to this address: 17K9SDPwvPCxmr3yM8BQcneF7d4FF9F5 . Application After your payment email me at f1rcuagm492@icamil.c
on with your server IP (164.90.229.254) and transaction ID and you will get a download link to your backup. Emails without transaction ID and server IP will be ignored. | 17K9SDPwvPCxmr3yM8BQcneF7d4FF9F5 |
+-----+-----+-----+-----+
```

Figura 37 Ataque a la base de datos

5. Implementar el uso de IAs para la generación de tareas, fases o metodologías dentro del proceso del DCU. Poder hablar con una IA para pedirle cual sería el siguiente paso para realizar dentro de nuestro proyecto.
6. Mejora en el sistema de generación de documentación automática, permitiendo el uso de imágenes, enlaces, tablas ...
7. Perfeccionar aquellos apartados que en la evaluación heurística se han visto peor cualificados, como, por ejemplo, informar a los usuarios de los diferentes errores que pueden provocar sus acciones.

Anexo 1 Manual del programador

En este documento se tratarán aquellos datos para que otro programador pueda identificar las partes más importantes de mi sistema y poder replicar mi aplicación y modificarla. Para ello es necesario explicar dos apartados, la estructura del sistema de ficheros y la forma correcta de lanzar mi aplicación una vez se tiene el proyecto base.

Estructura de ficheros

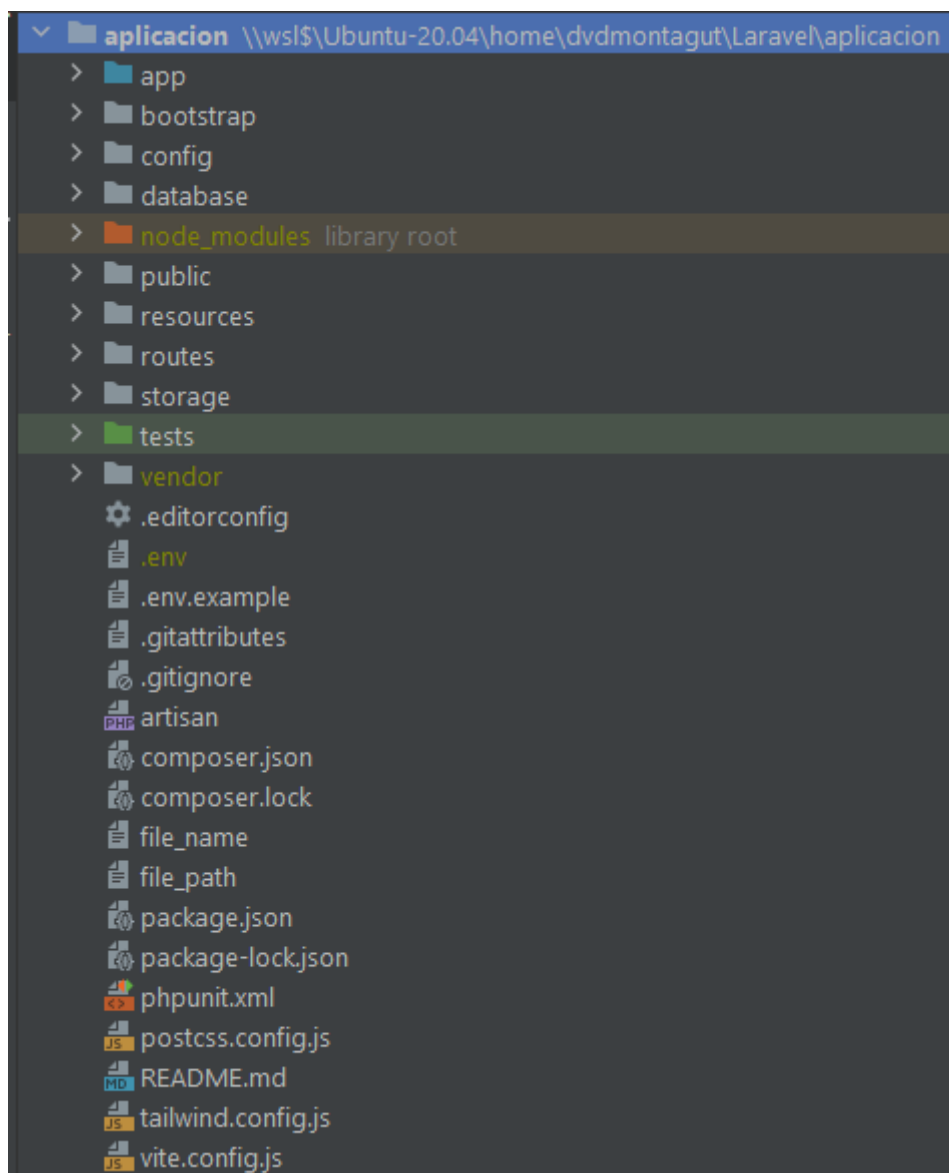


Figura 38 Estructura de ficheros

Para adquirir un mayor conocimiento sobre la estructura de directorios de Laravel se puede consultar su manual <https://laravel.com/docs/11.x/structure>

Pero en lo referente a este proyecto:

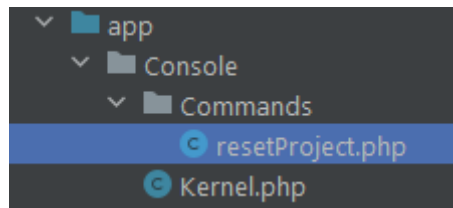


Figura 39 Fichero para los comandos

Dentro de app/Console/Commands podremos encontrar el comando resetProject que nos ayudará a poner todos los datos a 0 y reiniciar completamente el proyecto.

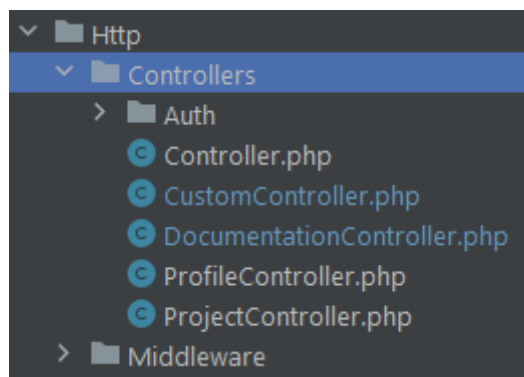


Figura 40 Fichero para los Controladores

Dentro de app/Http/Controllers encontramos los controladores que son los que se comunicarán con nuestra base de datos y les proporcionarán los datos a nuestras vistas. Auth es autogenerado y el resto han sido creados para solventar distintos problemas.

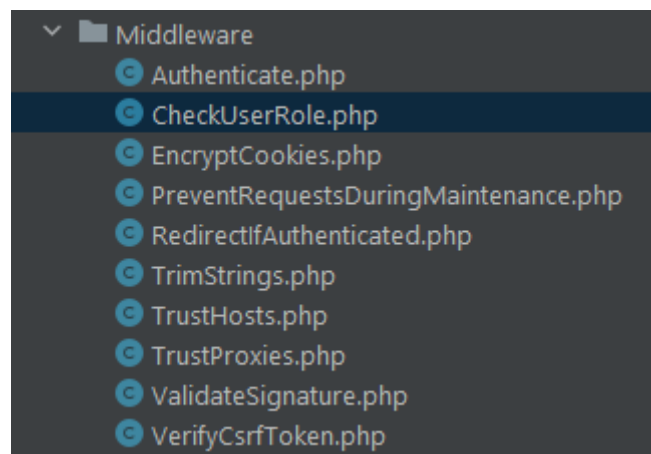


Figura 41 Fichero para el Middleware

Dentro de app/Http/Middleware encontraremos código que nos ayudará a procesar las peticiones y decir si permitimos el acceso o no, la mayoría son autogenerados pero hay excepciones como el CheckUserRole.php

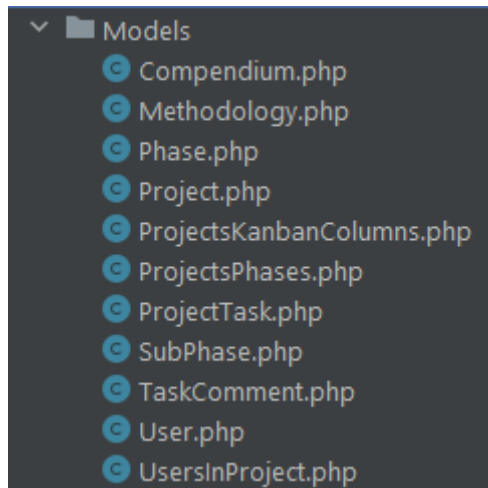


Figura 42 Fichero para los modelos

Dentro de app/Http/Models se encuentran todos los modelos empleados en la aplicación este directorio tiene mucha relación con las entidades de la base de datos.

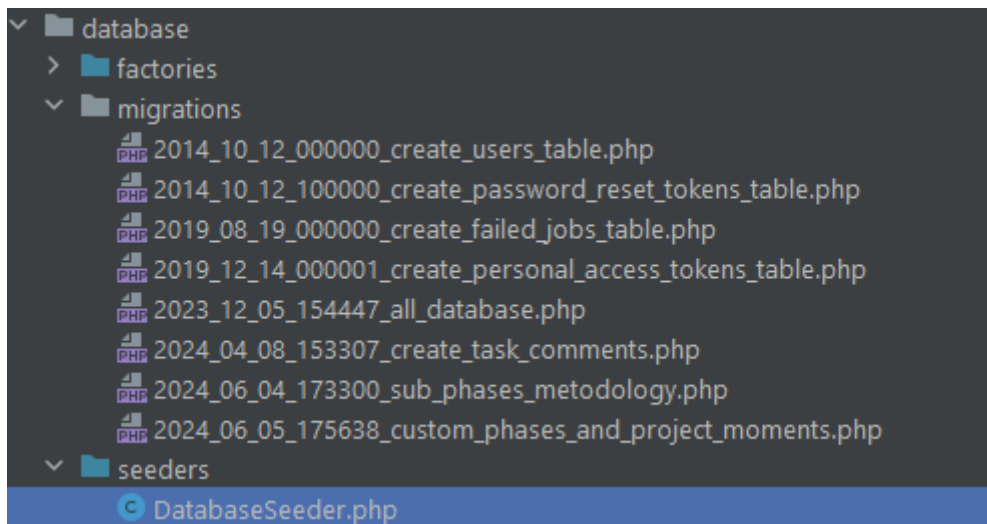


Figura 43 Ficheros para las migraciones y los seeders

Dentro de database encontramos los directorios de migrations y de seeders. Las migraciones nos ayudan a crear la base de datos en nuestro Docker de MariaDB. Y el Seeder nos ayuda a poner los valores por defecto que queremos que estén siempre en nuestra base de datos.

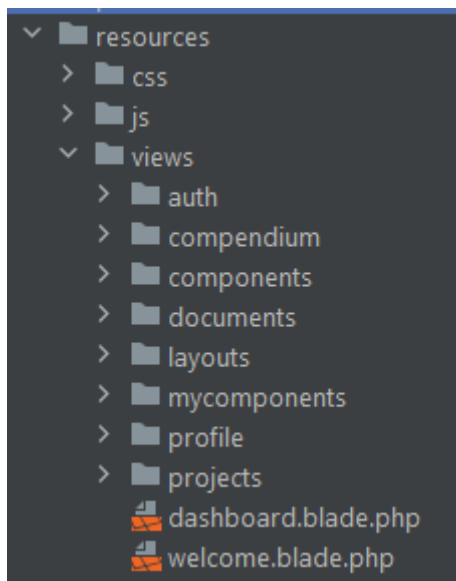


Figura 44 Fichero para las vistas

Dentro de resources/views tenemos todas las vistas empleadas en nuestro proyecto ordenadas en carpetas dependiendo de su función.

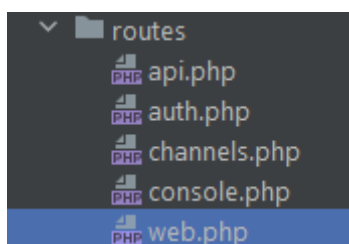


Figura 45 Fichero para las rutas

Por último, dentro de routes/web.php encontraremos todas las URIs necesarias para poder usar el proyecto, es decir en este archivo encontraremos especificaciones de cada ruta implementada en el proyecto, el tipo de petición que es (GET, POST) y a que controlador hacen referencia..

Pasos para hacer la instalación del proyecto

Para poder instalar el proyecto en un sistema con Ubuntu 20.04 hay que seguir los siguientes pasos:

Actualizar el sistema

```
apt-get update
apt-get upgrade
apt-get install git
apt-get purge php*.*
apt-get autoremove
apt-get autoclean
```

Instalar PHP y Docker

```
add-apt-repository ppa:ondrej/php
apt-get update
sudo apt install php8.3
sudo apt install apt-transport-https ca-certificates curl software-properties-common
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
sudo apt update
sudo apt install docker-ce
sudo systemctl start docker
sudo systemctl enable docker
```

Crear Docker MariaDB

```
sudo mkdir -p /var/lib/mysql
sudo chown -R $USER:$USER /var/lib/mysql
docker run -d \
  --name mariadb \
  -e MYSQL_ROOT_PASSWORD=contraseñaAElegir \
  -v /var/lib/mysql:/var/lib/mysql \
  -p 3306:3306 \
  mariadb:10.3

docker exec -it mariadb mysql -u root -p
# Luego ingresa la contraseña
CREATE DATABASE aplicacion;
exit
```

Instalar Composer

```
sudo apt install curl php-cli php-mbstring unzip
curl -sS https://getcomposer.org/installer | php
sudo mv composer.phar /usr/local/bin/composer
```

Clonar el repositorio o descargarlo

```
git clone https://github.com/dvdmontagut/TFM.git
```

Instalar node

```
nmp install
sudo apt remove nodejs npm
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -
sudo apt install -y nodejs
npm install
```

Instalar paquetes de php

```
apt-get install php8.3-intl
apt-get install php8.3-gd
apt-get install php8.3-mysql
sudo apt install php-curl
sudo apt install php-xml
```

Activar las extensiones en php.ini

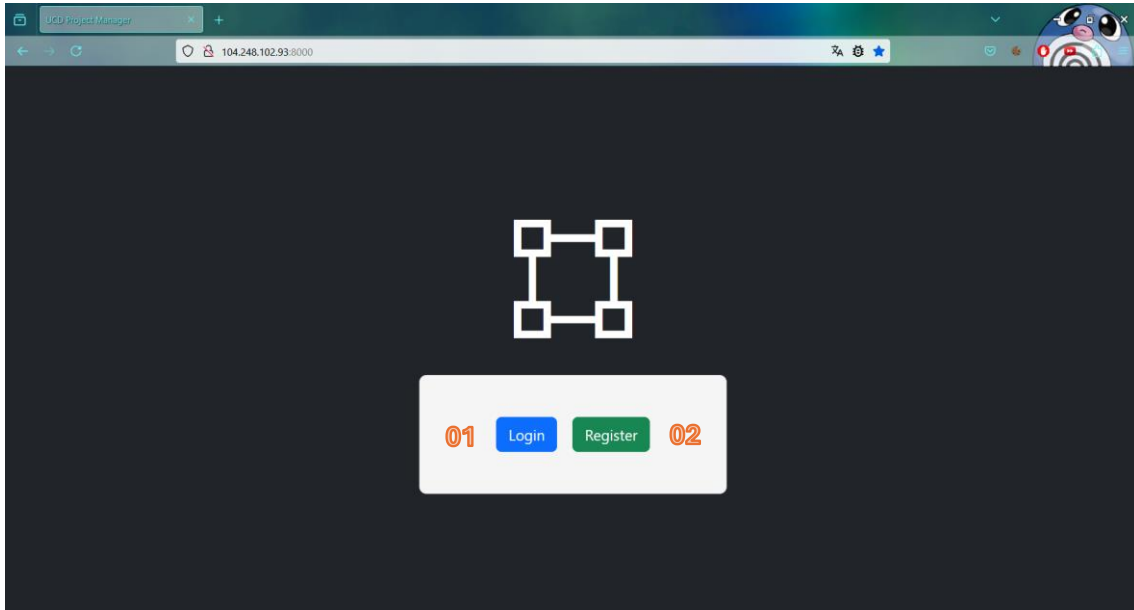
```
extension=curl
extension=fileinfo
extension=gd
extension=intl
extension=mbstring
extension=pdo_mysql
systemctl start apache2
```

Lanzar el proyecto

```
composer update
composer install
crear el .env
sudo ufw allow 8000
php artisan serve --host=0.0.0.0 --port=8000
```

Anexo 2 Manual de usuario.

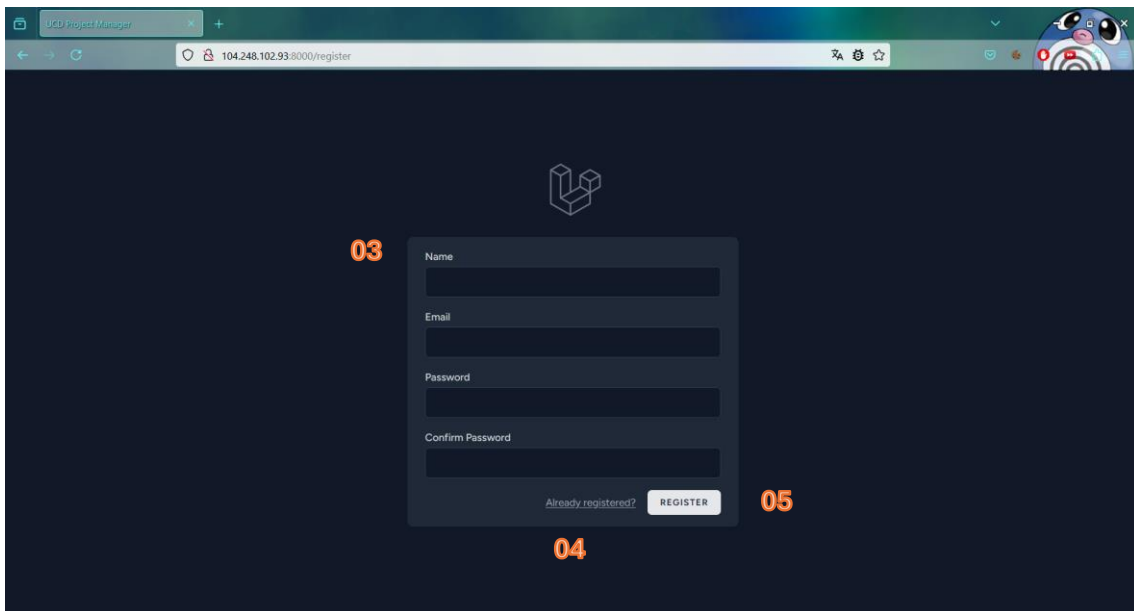
En este manual se explicará de forma breve la funcionalidad de cada botón o formulario que se encuentra en la aplicación web.



Manual de usuario 1

Botón para acceder al Login de la aplicación. (01)

Botón para acceder al Registro de la aplicación. (02)

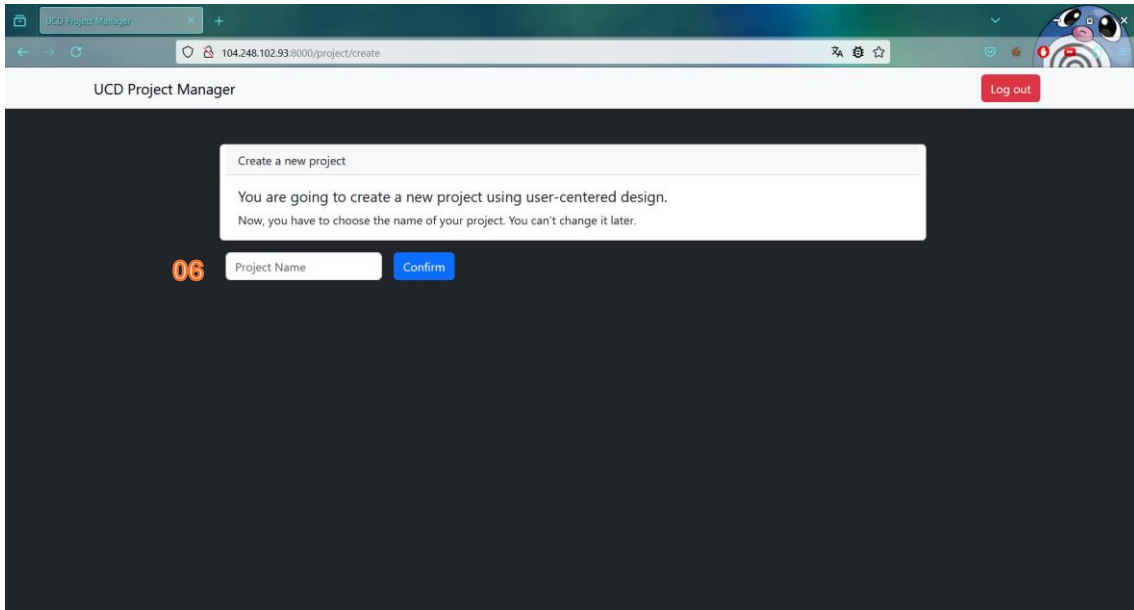


Manual de usuario 2

Formulario de acceso a la aplicación. (03)

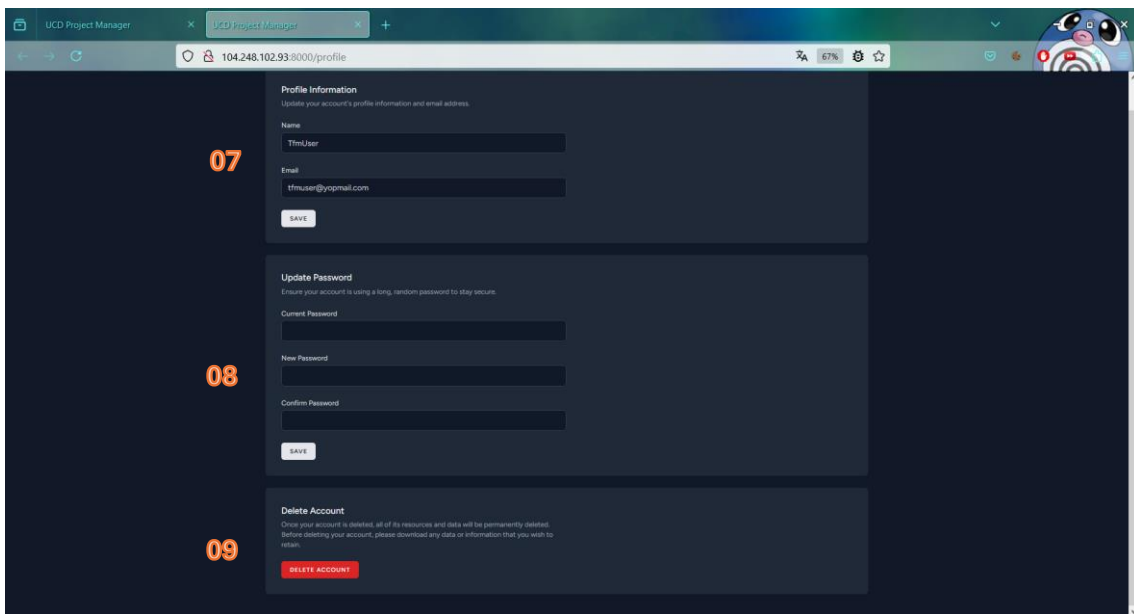
Botón para redirigir al Login. (04)

Botón de registro. (05)



Manual de usuario 3

Formulario para crear un nuevo proyecto. (06)

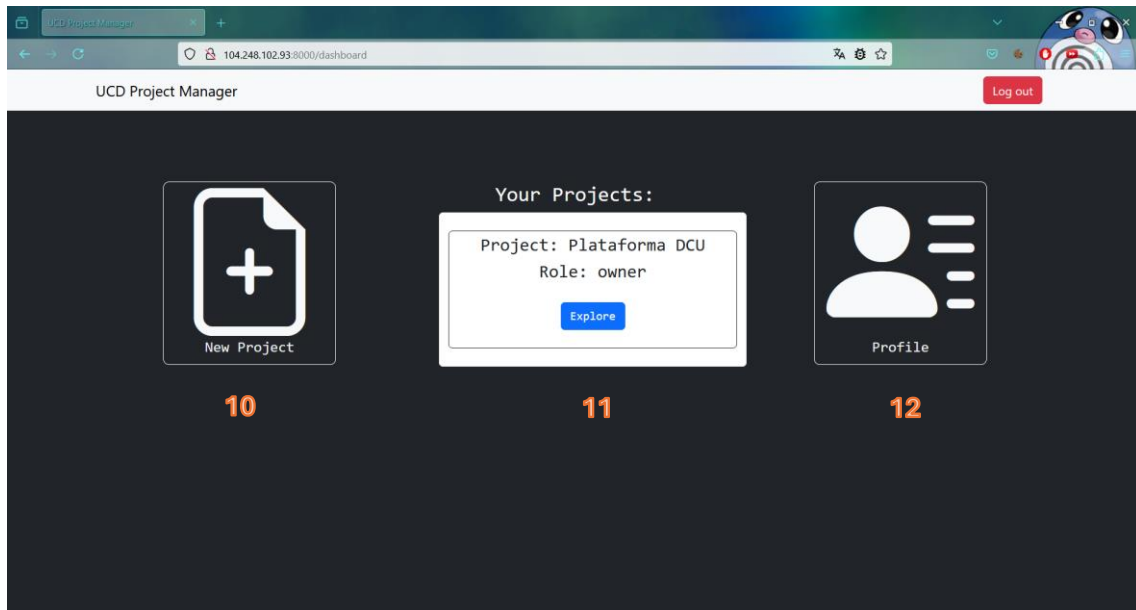


Manual de usuario 4

Formulario para cambiar la información de usuario. (07)

Formulario para cambiar la contraseña. (08)

Botón para eliminar el usuario. (09)

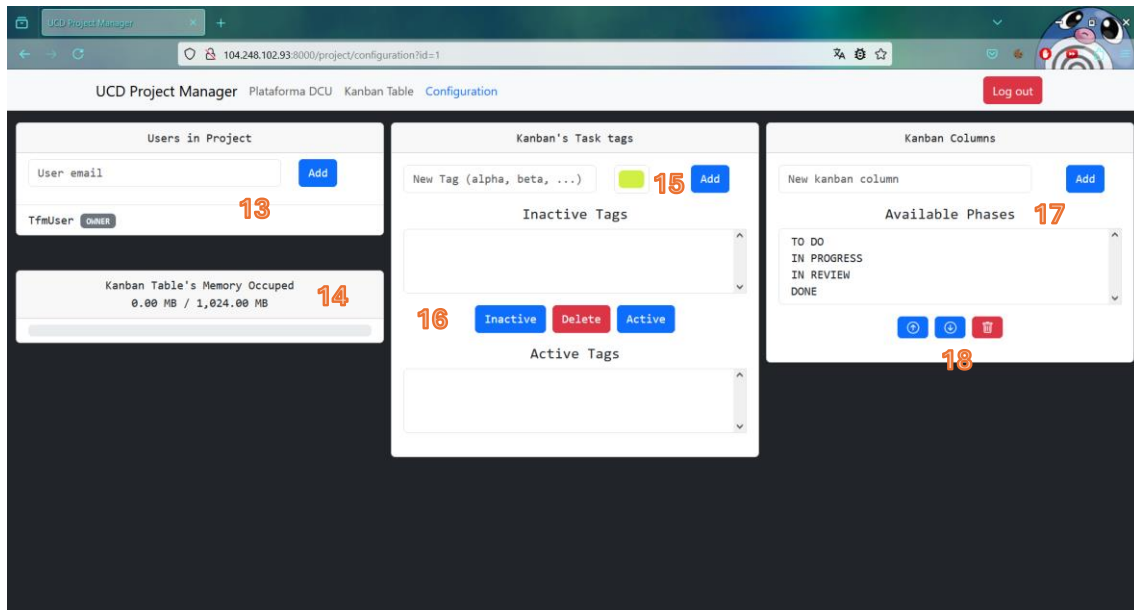


Manual de usuario 5

Botón para crear un nuevo proyecto. (10)

Menú de exploración de los proyectos activos. (11)

Botón para acceder al perfil del usuario. (12)



Manual de usuario 6

Formulario para añadir nuevos usuarios, es necesario que el usuario esté previamente registrado. (13)

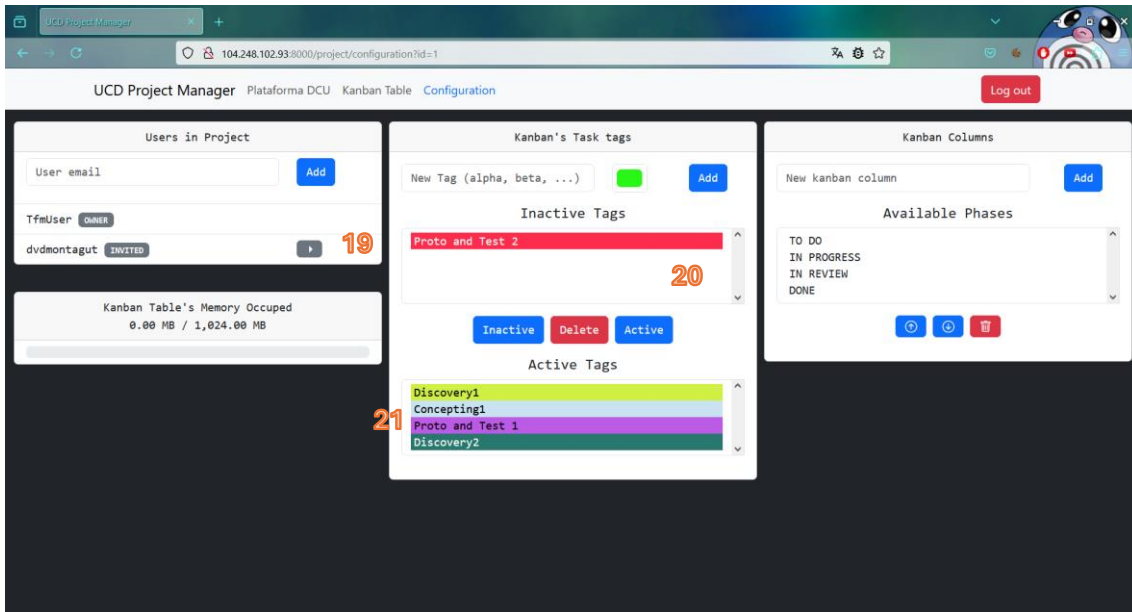
Memoria total usada por el proyecto. (14)

Formulario para añadir un nuevo tag al proyecto. (15)

Botones para desactivar, borrar o activar los tags para que sean visibles por el resto de los usuarios del proyecto. (16)

Formulario para añadir una columna a la tabla kanban. (17)

Botones para colocar o eliminar las columnas de la tabla kanban. (18)

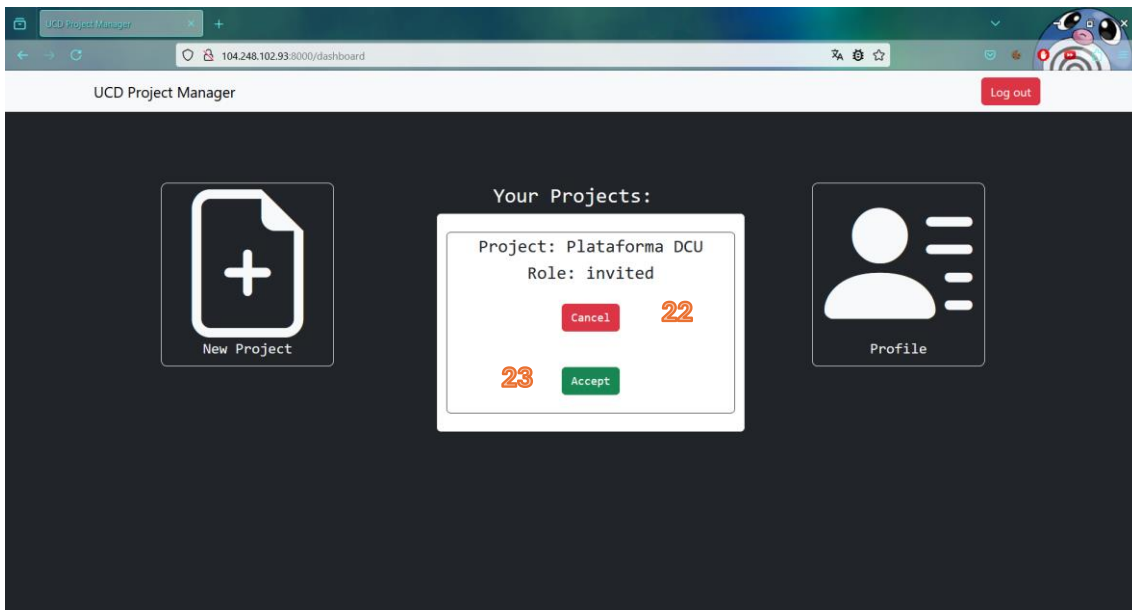


Manual de usuario 7

Botón para cambiar los roles o eliminar a un usuario del proyecto. (19)

Tags que aún están inactivas, estas no se mostrarán al resto de usuarios. (20)

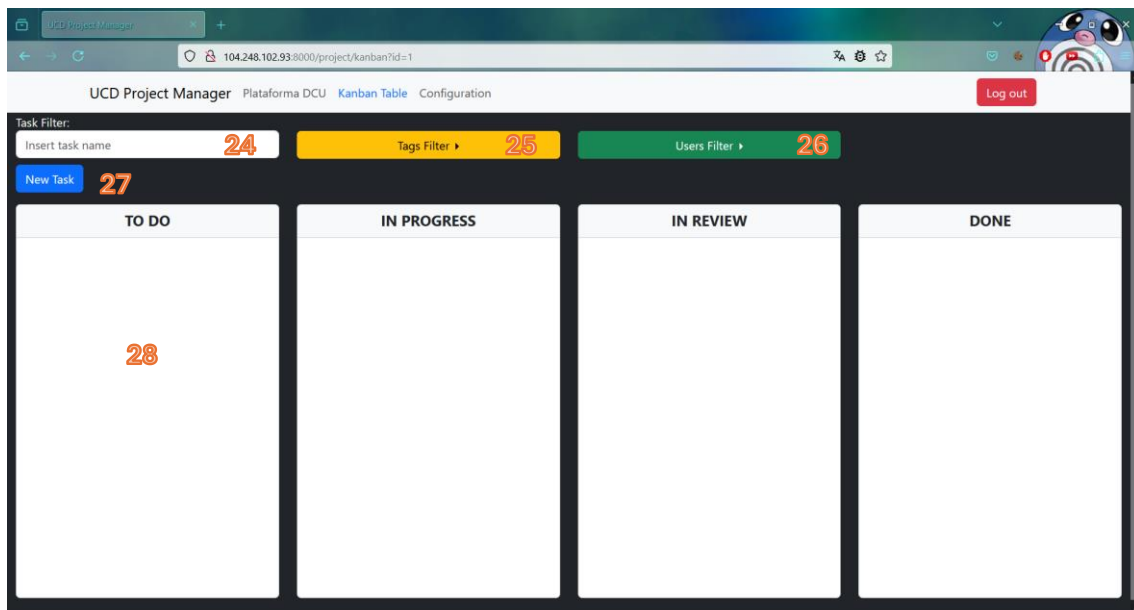
Tags que están activas y se mostrarán al resto de usuarios. (21)



Manual de usuario 8

Botón para rechazar la invitación a un proyecto. (22)

Botón para aceptar la invitación a un proyecto. (23)



Manual de usuario 9

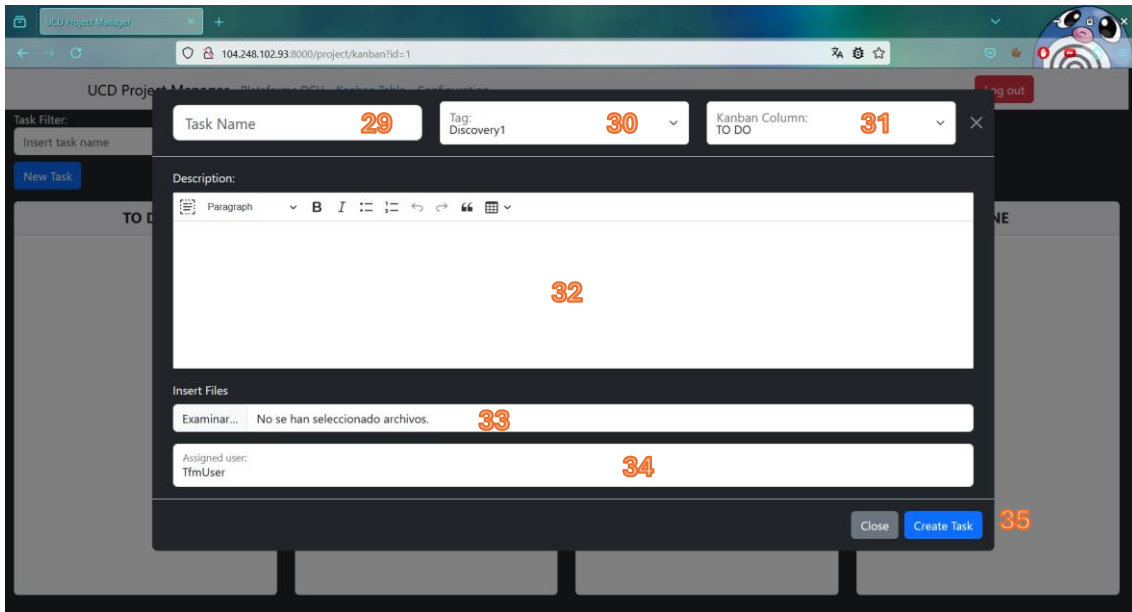
Filtro para buscar tareas por nombre. (24)

Filtro para buscar tareas por tag, los filtros son aditivos, todo lo que se seleccione se mostrará. (25)

Filtro para buscar tareas por tag, los filtros son aditivos, todo lo que se seleccione se mostrará. (26)

Botón para crear una nueva tarea. (27)

Tabla Kanban en esta tabla aparecen las tarjetas y se podrán deslizar entre las columnas. (28)



Manual de usuario 10

Nombre de la tarea. (29)

Tag asociado a la tarea. (30)

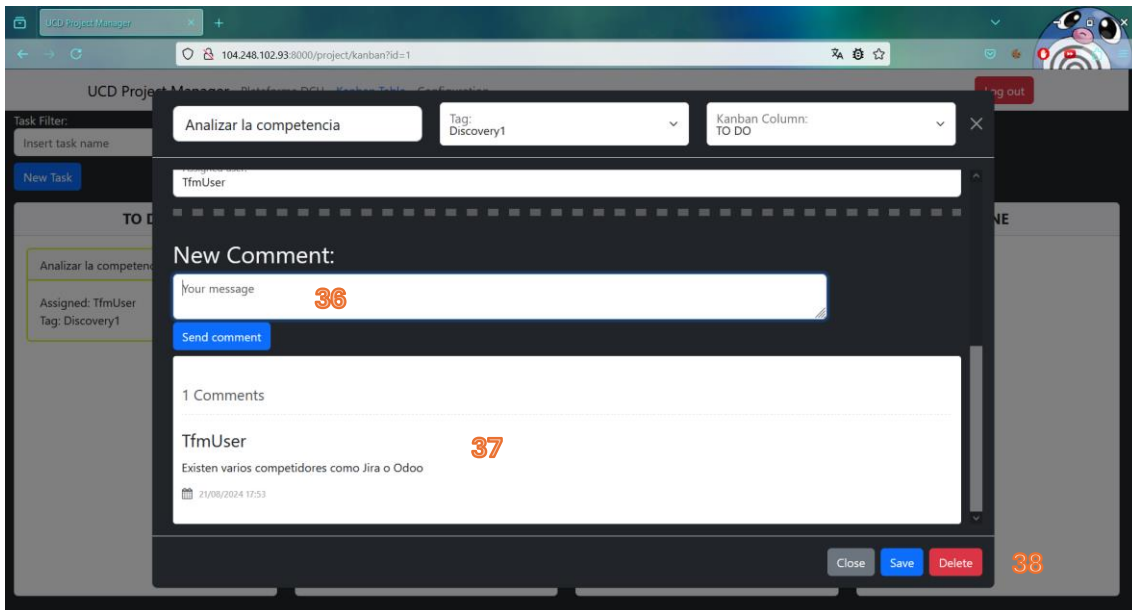
Estado de la tarea. (31)

Descripción de la tarea. (32)

Explorador de archivos para subir archivos a la tarea. (33)

Usuario asignado a la tarea. (34)

Botón de confirmación. (35)

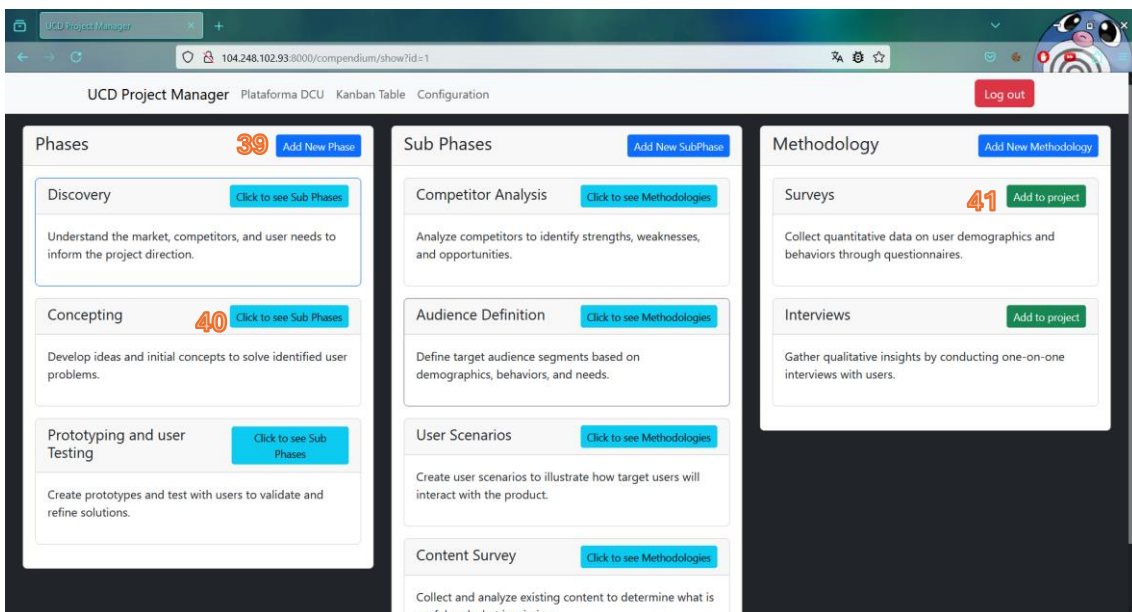


Manual de usuario 11

Formulario para añadir un nuevo comentario a la tarea. (36)

Historial de comentarios. (37)

Botón para modificar cambios o eliminar la tarjeta. (38)



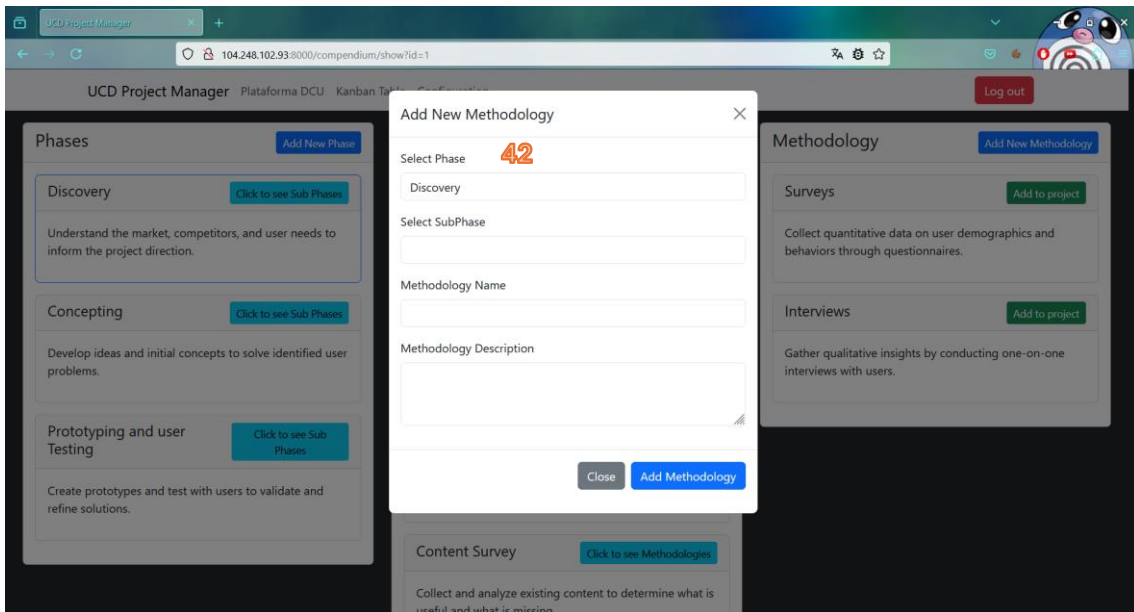
Manual de usuario 12

Botón para añadir una nueva fase, subfase o metodología. (39)

Botón para desplegar las subfases o metodologías de la fase o subfase seleccionadas.

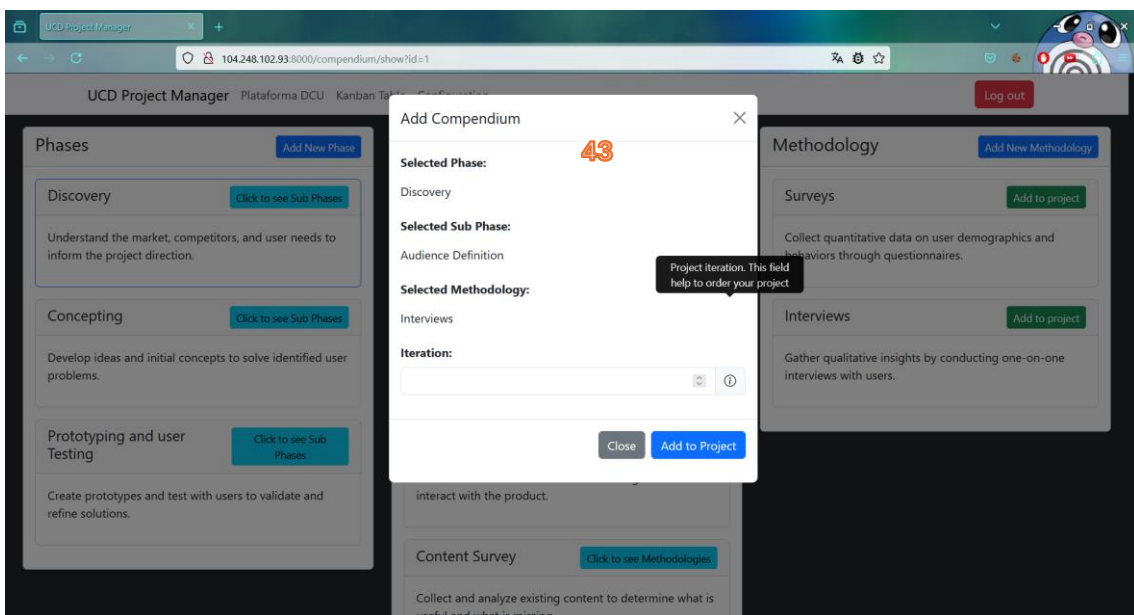
(40)

Botón para añadir una metodología al ciclo de proyecto. (41)



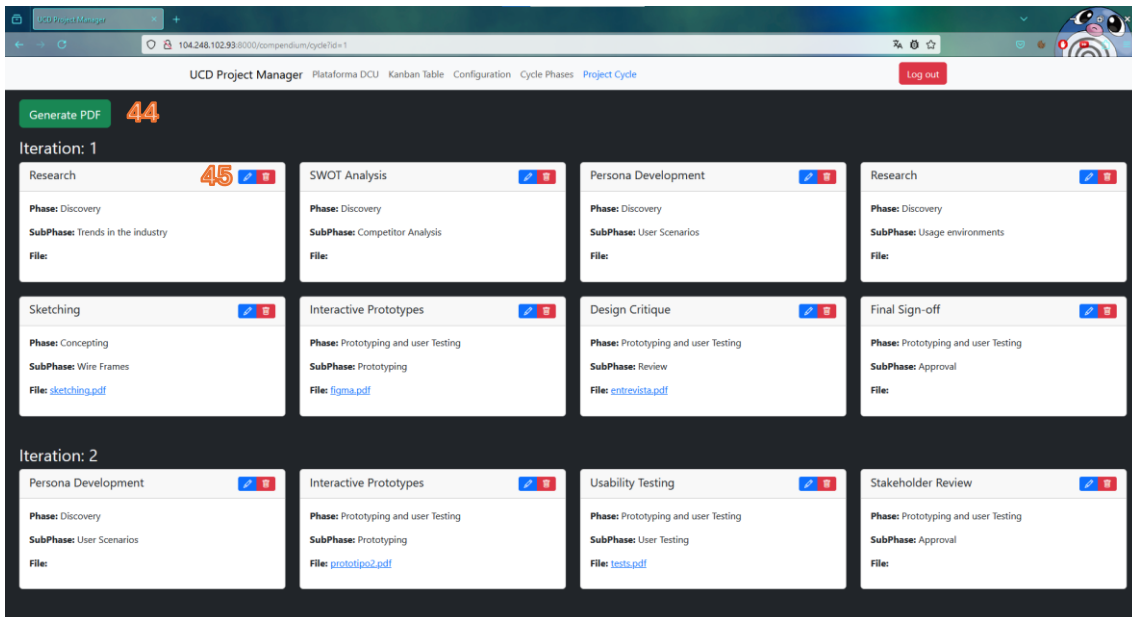
Manual de usuario 13

Formulario para añadir una nueva fase, subfase o metodología. (42)



Manual de usuario 14

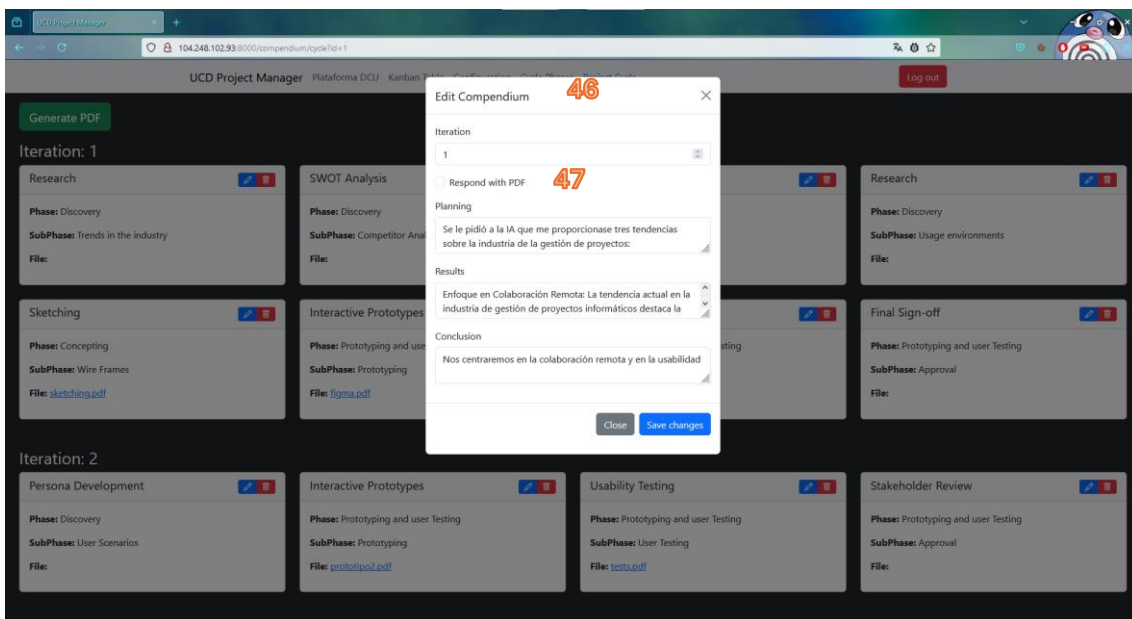
Formulario para añadir una metodología al ciclo. (43)



Manual de usuario 15

Botón para autogenerar la documentación del proyecto. (44)

Botones para editar o eliminar la metodología empleada en el proyecto. (45)



Manual de usuario 16

Formulario para describir lo ocurrido a lo largo de la metodología. (46)

Selector para elegir si responder mediante el formulario o mediante la subida de un archivo. (47)

Anexo 3 Documentación del proyecto generada por la propia aplicación.

En este apartado se mostrará una documentación generada con la aplicación creada. Decir que esta documentación es un resumen algo inexacto del contenido que se puede leer en la memoria y no es necesario leerlo, ya que el objetivo es ver el alcance que puede tener la autogeneración de documentación.

El archivo se llama **AutoDoc.pdf**

Anexo 4 Recursos y videos.

Entrevistas prototipo Figma

https://www.youtube.com/watch?v=FYpYZwW6x7s&ab_channel=DAVIDMONTAGUTPAMO

https://www.youtube.com/watch?v=TvzqmFaraP4&ab_channel=DAVIDMONTAGUTPAMO

https://www.youtube.com/watch?v=Jzj2XpN34Uc&ab_channel=DAVIDMONTAGUTPAMO

https://www.youtube.com/watch?v=8VV39dVlaA8&ab_channel=DAVIDMONTAGUTPAMO

[\[volver\]](#)

Primera versión de la aplicación

<https://youtu.be/AFj1BLh9yGw>

[\[volver\]](#)

Segunda versión de la aplicación

<https://youtu.be/zJKlkQjzfUk>

<https://youtu.be/DcJyaTqSCkk>

[\[volver\]](#)

Tercera versión de la aplicación

https://www.youtube.com/watch?v=omM1vZ3dWug&ab_channel=DAVIDMONTAGUTPAMO

[\[volver\]](#)

Entrevista *Affordance*

<https://youtu.be/ViV5ED6pcsU>

<https://youtu.be/Vc1jVg5IPgc>

[\[volver\]](#)

Bibliografía:

[1] Información sobre el Diseño Centrado en el Usuario:

Universidad Pontificia Bolivariana. (n.d.). *Diseño centrado en el usuario*. Recuperado de <https://repository.upb.edu.co/bitstream/handle/20.500.11912/6524/Dise%c3%b1o%20c%20en%20el%20usuario.pdf?sequence=1&isAllowed=y>

UNIR. (n.d.). *Diseño centrado en el usuario*. Recuperado de <https://www.unir.net/marketing-comunicacion/revista/disenio-centrado-usuario/>

(Interaction Design Foundation - IxDF. (2016). What is User Centered Design (UCD)? InteractionDesignFoundation-IxDF. Recuperado de <https://www.interaction-design.org/literature/topics/user-centered-design>)

[2] Ciclo del Diseño Centrado en el Usuario:

Hodder, D. (n.d.). *User-Centred Design Example*. Recuperado de <https://www.davidhodder.com/usercentredesignexample>

[3] Perfil de usuarios

Dam, R. F., & Teo, Y. S. (2024). Personas – A Simple Introduction. Interaction Design Foundation - IxDF. <https://www.interaction-design.org/literature/article/personas-why-and-how-you-should-use-them>

Documentación sobre Laravel:

Laravel. (n.d.). *Structure*. Recuperado de <https://laravel.com/docs/11.x/structure>