



Universidad de Jaén

Escuela Politécnica Superior de Jaén

Desarrollo de una herramienta para la trazabilidad de requisitos software

Autor: Juan Navarro Navarro

Grado: Ingeniería Informática

Director: Manuel José Barranco García
Departamento del director: Departamento de Informática

Fecha: Junio, 2024

Licencia CC



CREA

Agradecimientos

Quiero agradecer a todas las personas que durante estos años de carrera he conocido y han aportado algo positivo a mi vida: amigos, profesores, compañeros de piso, compañeros de clase, etc.

A mi tutor y profesor Manuel José, por ayudarme en este proyecto y estar tan comprometido conmigo en cada revisión y en cada reunión.

A mi familia, pero sobre todo a mi madre Rosa, a mi padre Pepe y a mi hermana. Gracias por todo vuestro esfuerzo y apoyo diario para que esto fuera posible, sin vosotros no lo habría conseguido.

A todos, GRACIAS.

ÍNDICE GENERAL

1.	INTRODUCCIÓN	6
1.1.	Objetivos del proyecto	7
1.1.1.	Objetivos	7
1.1.2.	Restricciones	8
1.2.	Objetivos del producto	8
1.2.1.	Objetivos funcionales	8
1.2.2.	Objetivos de calidad	9
1.2.3.	Restricciones	9
1.3.	Estructura del documento	10
2.	LA TRAZABILIDAD EN LA INGENIERÍA DE REQUISITOS	11
2.1.	Trazabilidad de requisitos	12
2.2.	Estudio de alternativas y estado del arte	13
2.2.1.	ReqView	13
2.2.1.1.	Características	14
2.2.1.2.	Ventajas	17
2.2.1.3.	Desventajas	17
2.2.1.4.	Conclusiones	17
2.2.2.	Spira Plan	17
2.2.2.1.	Características	18
2.2.2.2.	Ventajas	20
2.2.2.3.	Desventajas	20
2.2.2.4.	Conclusiones	21
2.2.3.	Accompa Requirements Management Software	21
2.2.3.1.	Características	22
2.2.3.2.	Ventajas	25
2.2.3.3.	Desventajas	25
2.2.3.4.	Conclusiones	25
2.2.4.	CodeBeamer	26
2.2.4.1.	Características	26
2.2.4.2.	Ventajas	28
2.2.4.3.	Desventajas	29
2.2.4.4.	Conclusiones	29
3.	PLANIFICACIÓN DEL PROYECTO	30
3.1.	Alcance del proyecto	31
3.2.	Metodología utilizada	32
3.3.	Planificación temporal	33
3.4.	Tecnologías utilizadas	35
3.5.	Valoración de la carga de trabajo requerida	37
3.6.	Presupuesto	47
3.6.1.	Coste hardware	47

3.6.2.	Coste software	48
3.6.3.	Coste total estimado	49
4.	ANÁLISIS	50
4.1.	Épicas y requisitos	51
4.1.1.	Épicas	51
4.1.2.	Requisitos	52
4.2.	Casos de uso	53
4.3.	Diagrama de casos de uso	54
4.4.	Actores del sistema	57
4.5.	Modelado Entidad/Relación	58
4.5.1.	Diagrama entidad/relación	58
5.	DISEÑO	60
5.1.	Arquitectura lógica	61
5.2.	Arquitectura física	61
5.3.	Diagrama de clases	63
5.4.	Diseño de la base de datos	65
5.4.1.	Diccionario de datos	65
5.5.	Diseño de la interfaz de usuario	71
6.	IMPLEMENTACIÓN	88
6.1.	Herramientas utilizadas	89
6.2.	Arquitectura de componentes software	89
6.3.	Detalles de la implementación	97
6.3.1.	Programación de la conexión con la base de datos	97
6.3.2.	Programación de la pantalla de registro e inicio de sesión	97
6.3.3.	Programación del menú lateral	101
6.3.4.	Programación del cuadro de usuario registrado	104
6.3.5.	Programación de la pantalla de creación de un proyecto	105
6.3.6.	Programación de la pantalla de creación y listado de requisitos	107
6.3.7.	Programación de la trazabilidad de cada uno de los requisitos	112
7.	PRUEBAS	114
7.1.	CP-01 Registro de usuario	115
7.2.	CP-02 Inicio de sesión	115
7.3.	CP-03 Cerrar sesión	116
7.4.	CP-04 Menú desplegable	116
7.5.	CP-05 Crear Proyecto	117
7.6.	CP-06 Modificar Proyecto	117
7.7.	CP-07 Eliminar Proyecto	118
7.8.	CP-08 Crear Grupo de trabajo	119
7.9.	CP-09 Visualizar Grupos de trabajo	119
7.10.	CP-10 Visualizar Proyectos	120
7.11.	CP-11 Crear Usuario	120
7.12.	CP-12 Visualizar usuarios	121
7.13.	CP-13 Crear y visualizar épicas	121

7.14.	CP-14 Crear y visualizar requisitos	122
7.15.	CP-15 Ver trazabilidad de requisitos	123
7.16.	CP-16 Crear nuevos artefactos	123
8.	MANUALES	125
8.1.	Manual de instalación	126
8.2.	Manual de usuario	129
8.2.1.	Inicio de la aplicación	129
8.2.2.	Pantalla de registro e inicio de sesión	129
8.2.3.	Pantalla de bienvenida	131
8.2.4.	Menú principal y botón para cerrar sesión	132
8.2.5.	Creación, modificación y eliminación de proyectos	133
8.2.5.1.	Creación	133
8.2.5.2.	Modificación	133
8.2.5.3.	Eliminación	134
8.2.6.	Listado de proyectos creados	136
8.2.7.	Creación y listado de grupos de trabajo	137
8.2.8.	Creación y listado de usuarios	139
8.2.9.	Ver información del proyecto	140
8.2.10.	Pantalla de épicas	141
8.2.11.	Pantalla de requisitos	143
8.2.12.	Trazabilidad e información del requisito	144
9.	CONCLUSIÓN FINAL	148
10.	BIBLIOGRAFÍA Y REFERENCIAS	149

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Inicio de ReqView.....	14
Ilustración 2: Manejo de proyectos en ReqView	18
Ilustración 3: Trazabilidad de requisitos en ReqView	16
Ilustración 4: Crear proyecto en Spire Plan	18
Ilustración 5: Manejo de proyectos en Spire Plan.....	19
Ilustración 6: Trazabilidad en Spire Plan	20
Ilustración 7: Manejo de proyectos en Accompa.....	22
Ilustración 8: Manejo de requisitos en Accompa.....	23
Ilustración 9: Trazabilidad en Accompa	24
Ilustración 10: Creación de proyectos en codeBeamer.....	26
Ilustración 11: Creación de requisitos en codeBeamer.....	27
Ilustración 12: Manejo de requisitos en codeBeamer.....	27
Ilustración 13: Gestión de usuarios en codeBeamer	28
Ilustración 14: Desarrollo iterativo incremental	32
Ilustración 15: Visual Studio Code.....	35
Ilustración 16: Lenguaje de programación	36
Ilustración 17: Pantalla XAMPP	36
Ilustración 18: Diagrama de casos de uso	56
Ilustración 19: Actores del sistema.....	57
Ilustración 20: Diagrama entidad/relación.....	59
Ilustración 21: Arquitectura física.....	62
Ilustración 22: Diagrama de clases	64
Ilustración 23: Contenido carpeta TFG.....	90
Ilustración 24: Contenido carpeta CSS.....	90
Ilustración 25: Contenido carpeta IMG	92
Ilustración 26: Contenido carpeta JS.....	92
Ilustración 27: Contenido carpeta PHP.....	93
Ilustración 28: Contenido para el backend	95
Ilustración 29: Clase “conexión_be.php”	96
Ilustración 30: Clase “inicioSesion.php”	98-99
Ilustración 31: Clase “login_usuario_be.php”	99
Ilustración 32: Clase “registro_usuario_be.php”.....	102
Ilustración 33: Programación menu lateral	101-103
Ilustración 34: Programación cuadro de usuario resgistrado.....	104
Ilustración 35: Clase “crear_proyecto.php”	105
Ilustración 36: Clase “registro_proyectos_bd.php”	106-107
Ilustración 37: Clase “requisitos.php”	108-110
Ilustración 38: Clase “registro_requisitos.php”	111
Ilustración 39: Clase “ver_requisito.php”	112-113
Ilustración 40: XAMPP iniciado	127
Ilustración 41: Base de datos phpMyAdmin.....	127
Ilustración 42: Pantalla de inicio de la aplicación	129

Ilustración 43: Pantalla de registro	130
Ilustración 44: Pantalla de inicio de sesión	130
Ilustración 45: Pantalla de bienvenida	131
Ilustración 46: Menú principal	132
Ilustración 47: Botón para cerrar sesión	132
Ilustración 48: Pantalla creación de proyectos	133
Ilustración 49: Pantalla con botón para modificación de proyectos.....	134
Ilustración 50: Pantalla modificación de proyectos.....	134
Ilustración 51: Pantalla eliminación de proyectos	135
Ilustración 52: Alerta de cambio	135
Ilustración 53: Pantalla con listado de proyectos creados.....	136
Ilustración 54: Pantalla de creación de grupos de trabajo	137
Ilustración 55: Pantalla con listado de grupos de trabajo creados.....	138
Ilustración 56: Pantalla con información del grupo de trabajo	138
Ilustración 57: Pantalla de creación de usuario	139
Ilustración 58: Pantalla con listado de usuarios creados.....	140
Ilustración 59: Pantalla de información de proyecto	141
Ilustración 60: Pantalla de creación de épicas.....	142
Ilustración 61: Pantalla con listado de épicas creadas.....	142
Ilustración 62: Pantalla de creación de requisitos	143
Ilustración 63: Pantalla con listado de requisitos creados.....	144
Ilustración 64: Pantalla con información del requisito	144
Ilustración 65: Cuadro de cambio de estado del requisito	145
Ilustración 66: Formulario de creación de artefactos	146
Ilustración 67: Cuadro con la trazabilidad del requisito	147

1. INTRODUCCIÓN

Hoy en día, los proyectos de ingeniería de software deben su éxito a la diversidad de herramientas que permiten el almacenamiento ordenado y accesible de la información generada durante el desarrollo de estos. Las "Herramientas de gestión de requisitos" o "Requirement management tools" (RMT) ocupan un lugar destacado en este tipo de herramientas, encargándose de la gestión de los requisitos desde su origen hasta su implantación en un sistema informático.

Hace algunos años la creación de aplicaciones software solía ser compleja y caótica al no seguir procesos estructurados, tomándose así las decisiones de manera improvisada. Esto conllevaba a problemas como falta de claridad y visibilidad, riesgo de omisión de información importante, dificultad para realizar cambios y problemas de comunicación, entre muchos otros.

Todos estos problemas fueron los que ocasionaron el surgimiento de las RMT, las cuales, tomando como punto de partida las necesidades de los usuarios, son capaces de almacenar la información, obtener una estructura organizada, facilitar la comunicación efectiva y servir de soporte para el desarrollo de una aplicación o sistema que garantice la satisfacción del cliente al cumplir con sus requisitos y expectativas.

1.1. Objetivos del proyecto

Exponemos a continuación los objetivos y restricciones del proyecto: qué esperamos conseguir como resultado de nuestro trabajo y cuáles son nuestras limitaciones que hemos de considerar en dicho proceso de desarrollo.

1.1.1. Objetivos

El objetivo de este trabajo es desarrollar una herramienta para la trazabilidad de requisitos software, que permita a los equipos de desarrollo gestionar eficientemente el ciclo de vida de los requisitos desde su adquisición hasta su implementación y validación, que aporte una mejora perceptible en la organización y seguimiento de los requisitos en comparación con métodos manuales o poco estructurados. Dividiremos el objetivo principal en varios objetivos más concretos:

- Estudio del estado del arte. Revisión bibliográfica y de herramientas existentes sobre la gestión de requisitos y trazabilidad.
- Estudio teórico que proporcione una base sólida para comprender los principios básicos de la trazabilidad de requisitos.

- Realización de un prototipo para la gestión y trazabilidad de requisitos, que permita la validación de los requisitos establecidos para este proyecto.
- Diseño e implementación de la herramienta de gestión y trazabilidad de requisitos.
- Desarrollo de la memoria del proyecto.

1.1.2. Restricciones

- **Limitaciones de recursos:** Los recursos disponibles, incluyendo tiempo y presupuesto, serán limitados y deberán ser administrados de manera eficiente.
- **Competencias técnicas:** El equipo de desarrollo (el alumno) deberá tener habilidades básicas en el desarrollo de software y diseño de interfaces de usuario.
- **Factores externos imprevistos:** Aunque el proyecto sea simple, pueden surgir imprevistos que afecten los plazos y la calidad del producto final, por lo que se deben planificar medidas de contingencia adecuadas.

1.2. Objetivos del producto

En este apartado se presentan los objetivos del producto. Se trata de una declaración de necesidades expresadas desde la perspectiva de los usuarios potenciales de la aplicación en desarrollo, diferenciando objetivos funcionales, de calidad y restricciones que afecten al producto final.

1.2.1. Objetivos funcionales

- Se desea disponer de una aplicación web, que incluya los elementos habituales (pantalla inicial, menús, footer, registro de usuario, inicio y fin de sesión, etc.)
- La aplicación deberá disponer de una base de datos para garantizar la permanencia de los mismos.
- La aplicación deberá permitir la gestión de proyectos: altas, bajas, consultas y modificación.
- La aplicación deberá ofrecer la gestión de grupos de trabajo, como un mecanismo para agrupar a los usuarios que trabajen de forma concurrente en proyectos.

- La aplicación permitirá trabajar con épicas, requisitos, casos de uso, casos de prueba y otros artefactos, gestionando la trazabilidad entre todos ellos.

1.2.2. Objetivos de calidad

- Diseñar una interfaz de usuario intuitiva y amigable que facilite la interacción de los usuarios con la herramienta de trazabilidad de requisitos.
- Flexibilidad o ajuste a diferentes contextos: La herramienta deberá ser lo suficientemente flexible como para adaptarse a diferentes entornos y necesidades de los usuarios finales.
- La implementación de la herramienta no requerirá tecnologías o conocimientos altamente especializados, permitiendo su desarrollo con recursos y habilidades disponibles en el contexto del proyecto.
- El estudio experimental demostrará la utilidad práctica de la herramienta en un entorno realista y sencillo, confirmando su efectividad para mejorar la gestión de requisitos.

1.2.3. Restricciones

Las restricciones del producto se imponen de manera anticipada y afectan directamente al producto final, no al proceso de desarrollo. Estas restricciones están justificadas por diversos motivos, tales como el costo, la facilidad de uso, la disponibilidad y el conocimiento previo del equipo de desarrollo.

- **Base de Datos:** El almacenamiento de los datos se realizará en una base de datos gratuita, específicamente MySQL, debido a su robustez, flexibilidad y amplia aceptación en la comunidad de desarrolladores.
- **Servidor Web:** El servidor web utilizado será Apache, conocido por su fiabilidad, flexibilidad y la amplia documentación y soporte que ofrece, integrado dentro del entorno XAMPP.
- **Lenguaje de Programación:** Al tener conocimientos previos y dominar la aplicación, se desarrollará en PHP para la lógica del lado del servidor, CSS para el diseño y la presentación de la interfaz de usuario, y JavaScript para la interactividad y la mejora de la experiencia del usuario.

- **Entorno de Desarrollo:** Se utilizará Visual Studio Code como entorno de desarrollo integrado (IDE), por su versatilidad, amplia gama de características y extensiones que facilitan la escritura de código en múltiples lenguajes.

1.3. Estructura del documento

El resto del documento se ha estructurado en los siguientes capítulos:

Cap. 2. La trazabilidad en la ingeniería de requisitos.

Se presenta el concepto de trazabilidad de requisitos y el estado del arte de las herramientas de gestión de requisitos que consideran la trazabilidad de los mismos.

Cap. 3. Planificación del proyecto.

Se describe la planificación del proyecto: alcance, metodología, plan temporal y plan de costes.

Cap. 4. Análisis.

Se describe el proceso de análisis que se ha llevado a cabo a lo largo del proyecto.

Cap. 5. Diseño.

Se presenta el diseño en el que se ha basado la implementación, incluyendo arquitectura, diagrama de clases, base de datos e interfaz de usuario.

Cap. 6. Implementación.

Se detalla el proceso de implementación del proyecto, abordando las diversas etapas y técnicas empleadas para convertir los requisitos definidos en un producto de software funcional.

Cap. 7. Pruebas.

Se realizan las pruebas finales de validación comprobando que el software produce los resultados correctos según los requisitos especificados.

Se concluye la memoria con el manual de instalación y manual de usuario, unas conclusiones finales y la bibliografía.

2. LA TRAZABILIDAD EN LA INGENIERÍA DE REQUISITOS

En este capítulo presentaremos el concepto de trazabilidad de requisitos con un mayor detalle y realizaremos un estudio del estado del arte en cuanto a herramientas de gestión de requisitos que contemplen la trazabilidad de los mismos.

2.1. Trazabilidad de requisitos ¹

La **ingeniería de requisitos** es una fase crucial en el desarrollo de software, ya que se encarga de definir, documentar y mantener los requisitos de un sistema.

Los **requisitos** son básicamente las necesidades y expectativas que los usuarios y otros interesados tienen respecto a un producto de software. Este proceso no solo es vital para asegurar que el producto final cumpla con las expectativas, sino que también ayuda a identificar posibles problemas desde el inicio del proyecto, lo que puede ahorrar tiempo y recursos a largo plazo.

Uno de los aspectos más importantes de la ingeniería de requisitos es la **gestión de requisitos**. Este proceso implica la identificación, organización y seguimiento de los requisitos a lo largo del ciclo de vida del proyecto. Una buena gestión de requisitos asegura que todas las partes involucradas entiendan claramente cuáles son las expectativas y cómo deben ser implementadas. Además, permite manejar los cambios en los requisitos de manera controlada y sistemática, evitando el caos que puede surgir cuando los cambios no se gestionan adecuadamente.

En este contexto, la **trazabilidad de requisitos** juega un papel fundamental. La trazabilidad se refiere a la capacidad de relacionar cada requisito con los elementos del proyecto que lo implementan y viceversa. Es como un hilo conductor que conecta cada requisito con su correspondiente diseño, desarrollo, pruebas y finalmente su implementación en el sistema.

La trazabilidad de requisitos permite responder a preguntas críticas durante el desarrollo de software, tales como:

- ¿Qué partes del sistema están afectadas por un cambio en un requisito?
- ¿Se han implementado todos los requisitos definidos por los interesados?
- ¿Se han verificado y validado todos los requisitos?
- ¿Cuál es el impacto de eliminar o modificar un requisito específico?

Para lograr una trazabilidad efectiva, es necesario contar con herramientas y métodos adecuados. **Las herramientas de gestión de requisitos** suelen proporcionar funcionalidades que permiten establecer y mantener estas relaciones de trazabilidad de manera automática. Por ejemplo, al crear un nuevo caso de uso, la herramienta puede enlazarlo automáticamente con los requisitos correspondientes, facilitando el seguimiento y la gestión.

El proceso de trazabilidad comienza con la **identificación de los requisitos** y la definición de sus relaciones con otros elementos del proyecto. Cada requisito debe estar claramente documentado y vinculado a sus fuentes, los artefactos asociados o el estado de este. A medida que el proyecto avanza, se deben mantener y actualizar estas relaciones para reflejar los cambios y asegurar que todos los elementos del proyecto están alineados con los requisitos originales.

En resumen, la trazabilidad en la ingeniería de requisitos no solo facilita el seguimiento y cumplimiento de los requisitos, sino que también proporciona una

visión clara del estado del proyecto en cualquier momento. Esto resulta en una mayor **calidad del software**, ya que permite detectar y corregir desviaciones de manera temprana, asegurando que el producto final cumpla con las expectativas y necesidades de los usuarios y otros interesados. La integración de buenas prácticas de trazabilidad en la gestión de requisitos es, por tanto, una inversión que puede traer beneficios significativos en términos de eficiencia y calidad del producto final.

2.2. Estudio de alternativas y estado del arte

El objetivo principal de esta fase es analizar detalladamente las opciones disponibles en el mercado, así como comprender el estado actual de la tecnología relacionada con el proyecto. Este análisis exhaustivo permite identificar las herramientas, lenguajes de programación, frameworks y plataformas que mejor se ajustan a los requisitos y objetivos específicos del proyecto.

El resultado de este estudio de alternativas y estado del arte servirá como base sólida para la selección y justificación de las tecnologías a utilizar en el desarrollo del sistema propuesto. Además, permitirá establecer un marco de referencia que oriente la toma de decisiones durante todas las etapas del proyecto, desde el diseño y la implementación hasta la evaluación y la entrega final.

2.2.1. ReqView ⁷

ReqView es una herramienta diseñada para ayudar a los equipos a organizar y gestionar los requisitos de sus proyectos de manera efectiva. Desde la captura inicial de requisitos hasta su seguimiento y gestión de cambios, ofreciendo una solución integral para garantizar la claridad y coherencia en todo el ciclo de vida del proyecto.

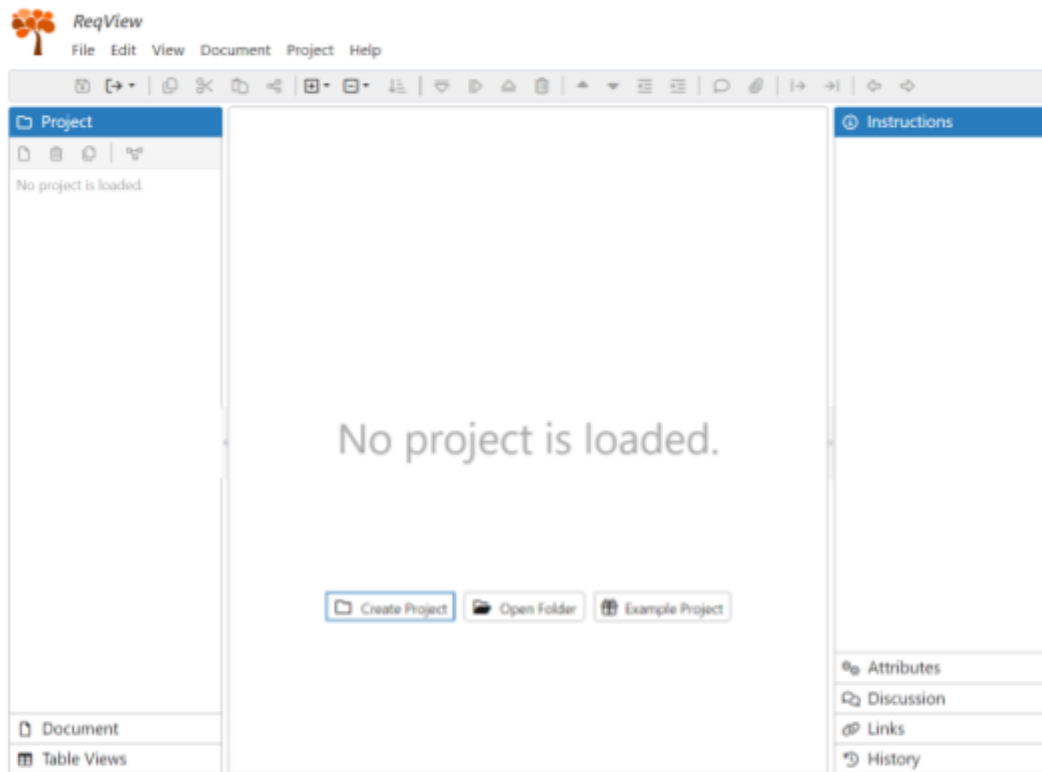


Ilustración 1: Inicio de ReqView

2.2.1.1. Características

Manejo de Proyectos

Con ReqView, los equipos pueden crear y organizar sus proyectos fácilmente. Pueden definir qué necesitan hacer y cómo van a hacerlo, y pueden ver cómo avanzan las cosas a lo largo del tiempo.

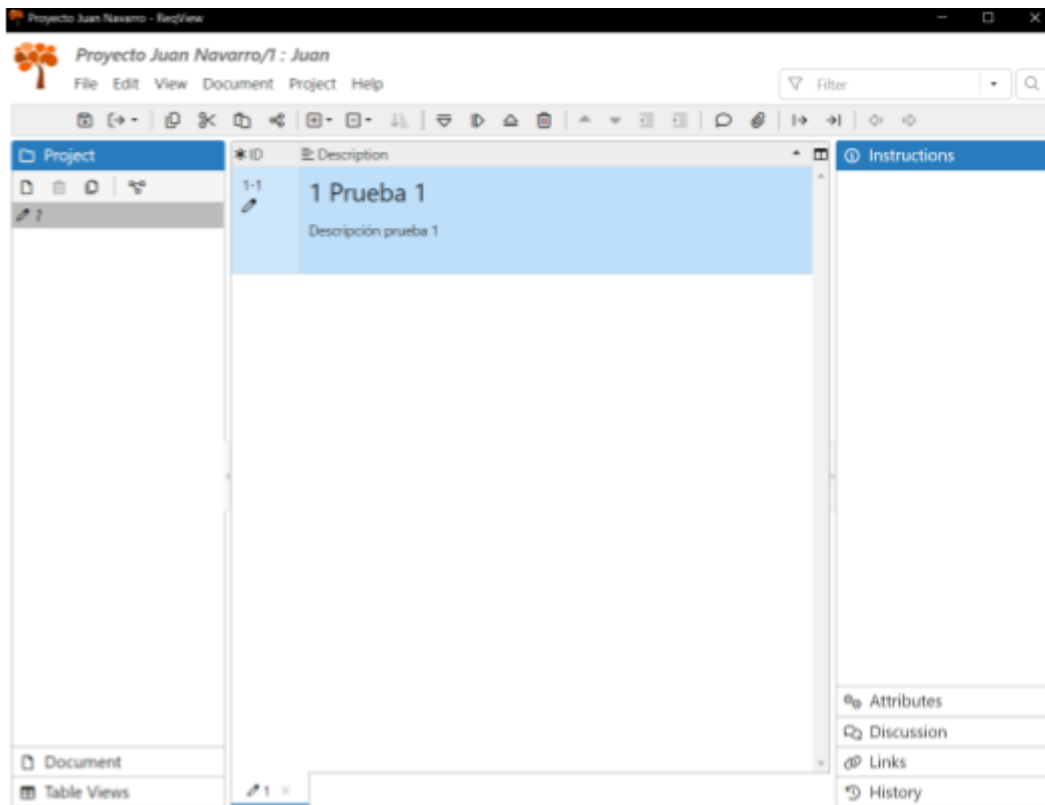


Ilustración 2: Manejo de proyectos en ReqView

Manejo de requisitos

ReqView permite a los equipos capturar y organizar los requisitos de manera estructurada y fácil de entender. Los usuarios pueden definir requisitos específicos, establecer relaciones entre ellos y organizarlos en una jerarquía clara para una mejor comprensión y trazabilidad.

Trazabilidad y seguimiento de cambios

Una característica destacada de ReqView es su capacidad para gestionar cambios en los requisitos de manera eficiente. Los equipos pueden realizar un seguimiento de las modificaciones realizadas en los requisitos, comparar versiones anteriores y establecer *baselines* para controlar y auditar los cambios a lo largo del tiempo.

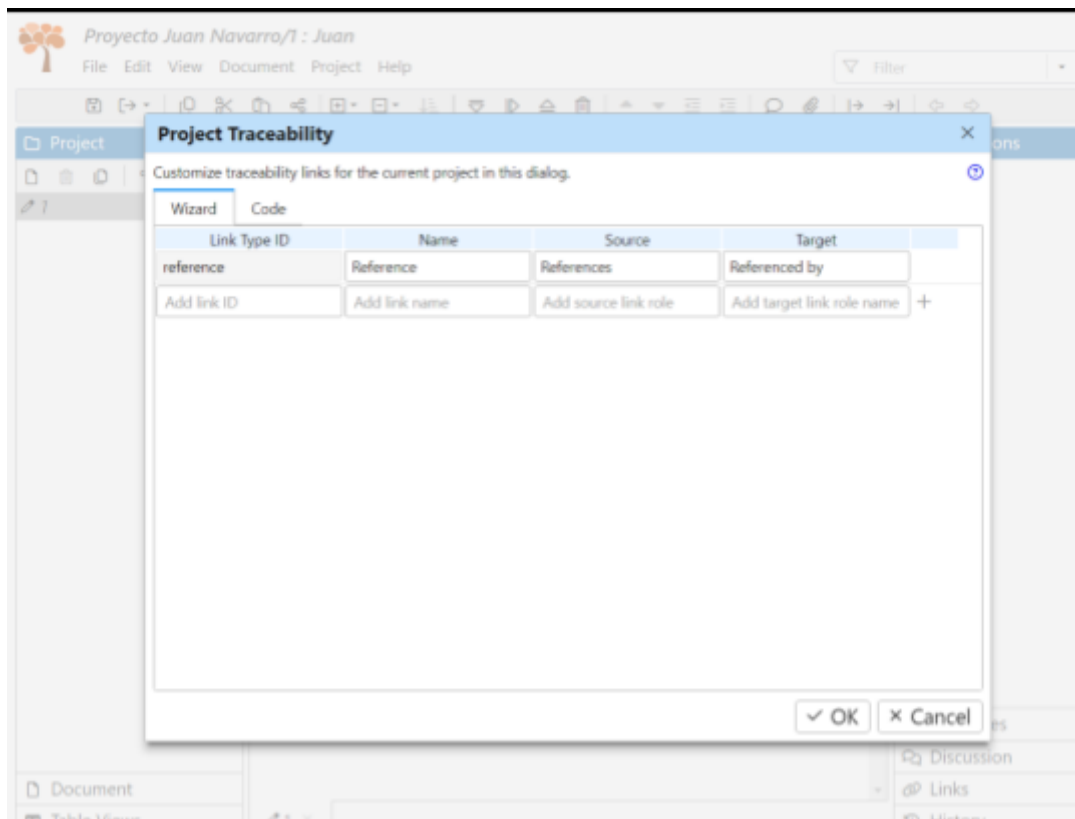


Ilustración 3: Trazabilidad de requisitos en ReqView

Colaboración y Comunicación

ReqView facilita la colaboración entre los miembros del equipo al proporcionar herramientas para revisar, comentar y aprobar requisitos. Los usuarios pueden trabajar juntos en la definición y validación de requisitos, lo que mejora la comunicación y evita malentendidos durante el desarrollo del proyecto.

Integración con Herramientas de Desarrollo

ReqView ofrece integraciones con una variedad de herramientas de desarrollo, como JIRA, Git y Microsoft Word, lo que permite a los equipos sincronizar y compartir información de manera eficiente entre diferentes plataformas y sistemas.

Generación de Informes Personalizados

La plataforma permite a los usuarios generar informes personalizados para evaluar el estado y la calidad de los requisitos. Los informes pueden incluir métricas de cumplimiento, trazabilidad de requisitos y análisis de impacto, proporcionando una visión completa del progreso del proyecto.

2.2.1.2. Ventajas

- Enfoque intuitivo y centrado en el usuario.
- Capacidades avanzadas de trazabilidad y gestión de cambios.
- Flexibilidad para adaptarse a diferentes metodologías de desarrollo.
- Soporte activo y actualizaciones regulares por parte del equipo de ReqView.

2.2.1.3. Desventajas

- Curva de aprendizaje inicial para nuevos usuarios.
- Requiere configuración y personalización para adaptarse a las necesidades específicas del proyecto.
- Algunas integraciones pueden ser limitadas o requerir ajustes adicionales.

2.2.1.4. Conclusiones

En conclusión, ReqView ofrece una solución completa y efectiva para la gestión de requisitos en proyectos de desarrollo de software y tecnológicos. Su enfoque intuitivo, capacidades avanzadas y flexibilidad lo convierten en una opción atractiva para equipos que buscan mejorar la claridad, coherencia y trazabilidad de sus requisitos.

2.2.2. Spira Plan ⁸

SpiraPlan es una plataforma integral de gestión de proyectos que ofrece una amplia gama de herramientas y funcionalidades diseñadas para facilitar la planificación, ejecución y seguimiento de proyectos de desarrollo de software y otros proyectos relacionados con la tecnología. Esta herramienta combina características de gestión de proyectos, gestión de requisitos, seguimiento de incidencias, gestión de pruebas y gestión de recursos en una sola solución integrada.

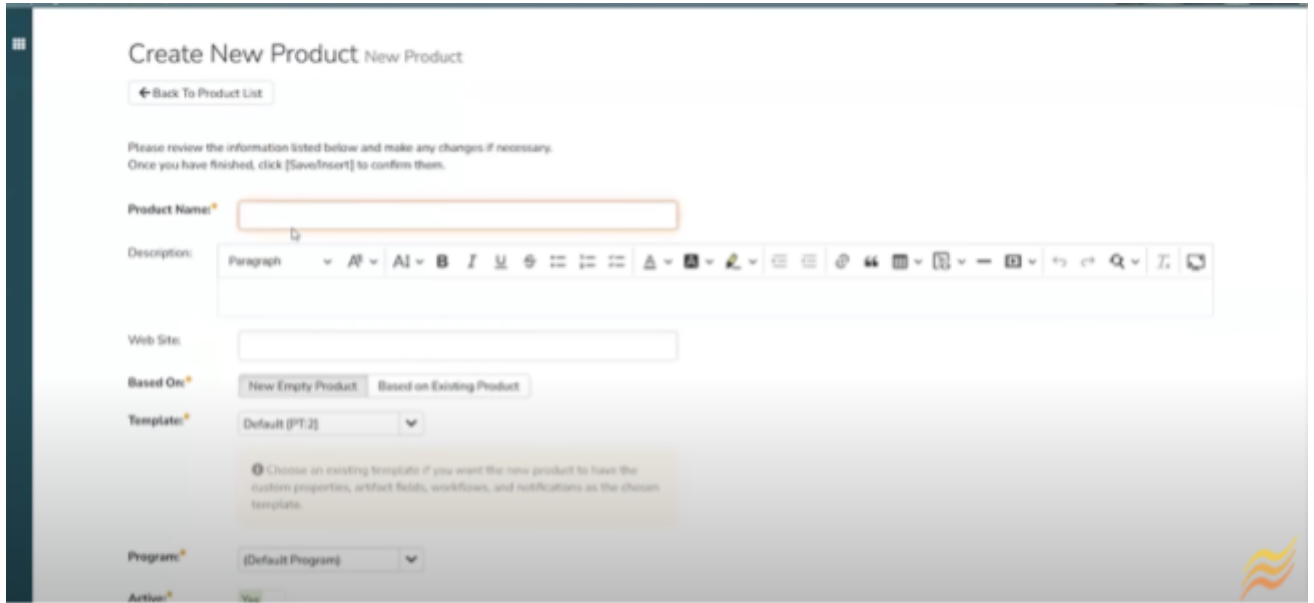


Ilustración 4: Crear proyecto en Spire Plan

2.2.2.1. Características

Manejo de proyectos

SpiraPlan proporciona una plataforma robusta para la gestión de proyectos, permitiendo la creación de cronogramas detallados, la asignación de tareas a miembros del equipo, y el seguimiento del progreso a través de tableros Kanban y gráficos de Gantt. Cada proyecto muestra la información sobre el grupo al que pertenece, si está activo o no y la fecha de creación.

View / Edit Products

The following products have been created in the system. To edit an existing product, please click on the 'Edit' link. To create a new product, click on the [Add] button.

[+ Add](#) [Filter](#) [X Clear Filter](#)

Product Name ▲▼	Program ▲▼	Creation Date ▲▼	Template ▲▼	Active ▲▼	ID ▲▼	Operations
Automated Driving System	Automotive	12-Nov-2023	Default	No	PR122	View Edit Clone Delete
Billing System	Back Office	12-Nov-2023	Regulated Industries	No	PR113	View Edit Clone Delete
Clinical Medical Devices	Clinical Trials	12-Nov-2023	Regulated Industries	No	PR119	View Edit Clone Delete
Company Website	Sales and Marketing	12-Nov-2023	Default	Yes	PR116	View Edit Clone Delete
Core Banking (T24)	Back Office	12-Nov-2023	Regulated Industries	No	PR112	View Edit Clone Delete
Customer Relationship Management (CRM)	Sales and Marketing	12-Nov-2023	Default	Yes	PR117	View Edit Clone Delete
Electronic Control Unit	Automotive	12-Nov-2023	Default	No	PR123	View Edit Clone Delete
ERP: Financials	Corporate Systems	12-Nov-2023	Default	Yes	PR114	View Edit Clone Delete
ERP: Human Resources	Corporate Systems	12-Nov-2023	Default	Yes	PR115	View Edit Clone Delete
ERP: Inventory Management	Inventory Systems	12-Nov-2023	Default	No	PR120	View Edit Clone Delete
ERP: Warehouse Management	Inventory Systems	12-Nov-2023	Default	No	PR121	View Edit Clone Delete
Flight Control System	Aviation Platform	12-Nov-2023	Regulated Industries	No	PR116	View Edit Clone Delete
Flight Planning System	Aviation Platform	12-Nov-2023	Regulated Industries	No	PR117	View Edit Clone Delete
Guidance System	Space Platform	12-Nov-2023	Regulated Industries	No	PR118	View Edit Clone Delete

Ilustración 5: Manejo de proyectos en Spira Plan.

Manejo de requisitos

En el contexto del desarrollo de software, la gestión de requisitos juega un papel fundamental en la definición y comprensión de los objetivos del proyecto. SpiraPlan facilita este proceso mediante la creación de historias de usuario, epics y otros artefactos de requisitos, que pueden ser organizados, priorizados y enlazados entre sí para proporcionar una visión clara y coherente del alcance del proyecto.

Seguimiento de incidencias

El seguimiento de incidencias es esencial para identificar, registrar y resolver problemas, errores y mejoras a lo largo del ciclo de vida del proyecto. SpiraPlan ofrece un sistema completo para gestionar estas incidencias, permitiendo su categorización, asignación a miembros del equipo, priorización y seguimiento hasta su resolución satisfactoria.

Gestión de pruebas

La calidad del software es un aspecto crítico en cualquier proyecto de desarrollo, y SpiraPlan facilita la gestión de pruebas mediante la creación y ejecución de casos de prueba, la asignación de pruebas a miembros del equipo, y el seguimiento de los resultados obtenidos. Esta funcionalidad garantiza una cobertura exhaustiva de pruebas y una detección temprana de posibles fallos o errores.

Gestión de recursos y trazabilidad

La optimización de recursos es esencial para maximizar la eficiencia y la productividad del equipo. SpiraPlan permite gestionar eficazmente los recursos humanos y materiales del proyecto, asignando tareas y actividades de manera equitativa. Sin embargo, Spira no ofrece una función de seguimiento como tal, sino que agrupa las necesidades de forma jerárquica sin tener en cuenta que una necesidad dependiente de una característica puede estar asociada a otra dependiente de otra característica.

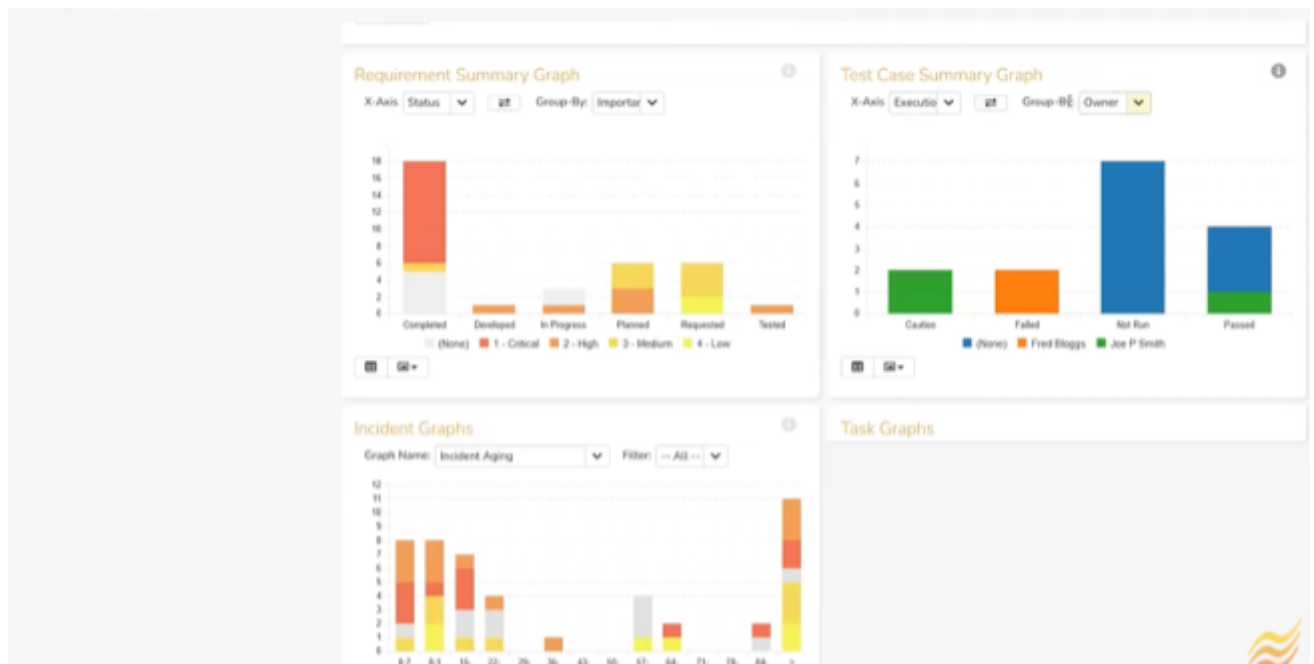


Ilustración 6: Trazabilidad en Spira Plan

2.2.2.2. Ventajas

Entre las ventajas y beneficios de SpiraPlan se incluyen su integración completa de funciones en una sola plataforma, su facilidad de uso y su capacidad para adaptarse a diferentes metodologías de desarrollo. Además, ofrece mejoras significativas en la colaboración, la comunicación y la productividad del equipo, así como una mayor visibilidad y control sobre el progreso del proyecto.

2.2.2.3. Desventajas

Sin embargo, algunas desventajas de SpiraPlan pueden incluir un periodo de aprendizaje inicial para los usuarios nuevos, la necesidad de mantener actualizada la información en la plataforma para garantizar su eficacia, y posibles limitaciones en

la personalización de ciertas funcionalidades según las necesidades específicas del proyecto o equipo.

2.2.2.4. Conclusiones

En resumen, permite a los equipos de desarrollo colaborar de manera efectiva, coordinar actividades, gestionar el ciclo de vida del proyecto, realizar un seguimiento detallado del progreso y asegurar la calidad del producto final, todo ello dentro de un entorno centralizado y fácil de usar. Esta plataforma es especialmente útil para equipos de desarrollo ágiles que buscan una solución completa para la gestión de sus proyectos de software.

Así pues, SpiraPlan emerge como una herramienta integral y poderosa para la gestión de proyectos en el ámbito de la ingeniería de software y la tecnología.

2.2.3. Accompa Requirements Management Software ⁹

Accompa Requirements Management Software es una plataforma especializada en la gestión de requisitos, diseñada para ayudar a los equipos de desarrollo a capturar, organizar y gestionar de manera efectiva los requisitos de sus proyectos. Ofrece una amplia gama de herramientas y funcionalidades centradas en simplificar el proceso de definición, seguimiento y trazabilidad de los requisitos, facilitando así la colaboración entre equipos y garantizando la entrega exitosa de productos de alta calidad.

2.2.3.1. Características

Manejo de proyectos

Accompa permite a los equipos gestionar proyectos de manera eficiente y sencilla, sin opciones de personalización y únicamente para metodologías de desarrollo tradicionales.. Además, facilita la integración de los requisitos del proyecto en el plan general, permitiendo una mejor alineación entre los objetivos del proyecto y los requisitos específicos.

ACCOMP

Start Features Use Cases **Requirements** Baselines AutoCapture

Dashboard Add Requirement Add/Edit Field Values View Requirements Popular Tags Discussions Import Export Create Document

Search: Go [Advanced Search](#) Logged in to: Remarkable

Add Requirement

Save Cancel Save & Add Another

Basic Information Click field labels for explanation.

Title: Ability to import XML files

Description: Several customers have asked us to support importing XML files. Customers include:

Company Name	Licenses	Account Size
ABC Inc.	27	\$\$
XYZ Corp.	71	\$\$\$
Awesome Widgets	5	\$

Products: RemarkableCalc [Remove](#) [Add Another](#)

Tags:

Status:

Ilustración 7: Manejo de proyectos en Accompa

Manejo de requisitos

La plataforma simplifica el proceso de gestión de requisitos al ofrecer herramientas para capturar, organizar y priorizar los requisitos del proyecto. Permite a los usuarios definir relaciones entre los requisitos, gestionar cambios y realizar un seguimiento de su estado a lo largo del ciclo de vida del proyecto.

The screenshot shows the 'Add Requirement' form in the Accompa application. The interface includes a top navigation bar with tabs for 'Start', 'Requirements', 'Features', and 'AutoCapture'. A user is logged in as 'Remarkable Software, Inc'. Below the navigation bar is a search bar and a 'Go' button. The main form area is titled 'Add Requirement' and contains several sections: 'Basic Information' (with a hint to click field labels for explanation), 'Title' (a text input field), 'Description' (a large text area with a 'Check Spelling' button and a 'Display' dropdown set to 10 lines), 'Products' (a text input field with a 'Select' button), 'Tags' (a text input field), 'Status' (a dropdown menu), 'Owner' (a dropdown menu), 'Priority' (a dropdown menu), 'Type of Requirement' (a dropdown menu), 'Needed By Date' (a text input field with a 'Select' button), 'Requested By' (a text input field), and 'Effort Estimate (Man-Days)' (a text input field). At the bottom of the form are three buttons: 'Save', 'Cancel', and 'Save & Add Another'.

Ilustración 8: Manejo de requisitos en Accompa

Trazabilidad

Accompa Requirements Management Software no ofrece una función de seguimiento como tal, sino que agrupa las necesidades de forma jerárquica.

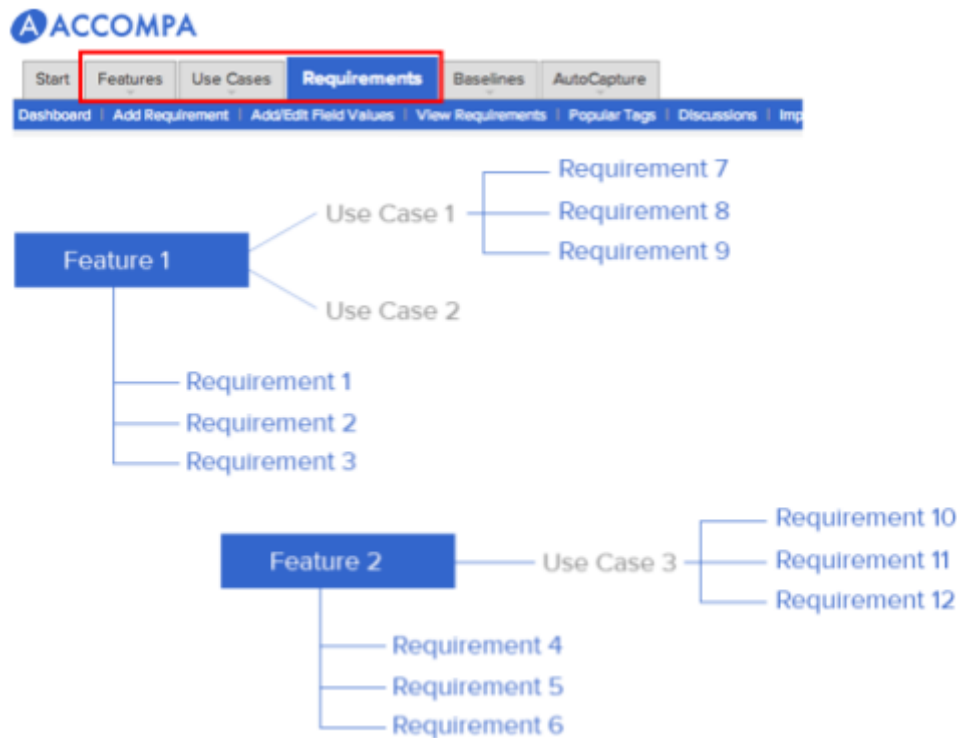


Ilustración 9: Trazabilidad en Accompa

Gestión de Pruebas

La plataforma facilita la gestión de pruebas al permitir la creación y ejecución de casos de prueba, así como el seguimiento de los resultados obtenidos. Esto asegura una cobertura completa de pruebas y una detección temprana de posibles problemas, contribuyendo así a la entrega de un producto de alta calidad.

Gestión de Recursos

Accompa ayuda a optimizar la utilización de recursos al permitir la asignación eficiente de tareas y actividades a los miembros del equipo. Además, ofrece herramientas para monitorizar la carga de trabajo de cada recurso y gestionar los recursos disponibles de manera efectiva.

Casos de Uso

En la práctica, Accompa ha demostrado su eficacia en una variedad de proyectos de desarrollo de software y tecnológicos. Existen otras opciones y herramientas como un historial de versiones que, además, permite volver a una versión anterior. Como parte de administración del software Accompa ofrece servicios REST y SOAP para comunicar la aplicación con otras.

2.2.3.2. Ventajas

Entre las ventajas de Accompa se incluyen su enfoque especializado en la gestión de requisitos, su facilidad de uso y su capacidad para adaptarse a las necesidades específicas de los equipos de desarrollo. Además, ofrece una mayor transparencia, colaboración y trazabilidad en el proceso de gestión de requisitos, lo que contribuye a la entrega exitosa de proyectos.

2.2.3.3. Desventajas

Sin embargo, algunas desventajas de Accompa pueden incluir una curva de aprendizaje inicial para los nuevos usuarios y posibles limitaciones en la personalización de ciertas funcionalidades según las necesidades específicas del proyecto o equipo.

2.2.3.4. Conclusiones

En conclusión, Accompa Requirements Management Software se posiciona como una herramienta integral y especializada en la gestión de requisitos, ofreciendo funcionalidades avanzadas para simplificar el proceso de captura, organización y seguimiento de los requisitos del proyecto. Si bien presenta algunas limitaciones, su enfoque centrado en la gestión de requisitos y sus ventajas en términos de transparencia, colaboración y trazabilidad lo convierten en una opción atractiva para equipos de desarrollo en busca de mejorar la eficiencia y la calidad en la entrega de sus proyectos.

2.2.4. CodeBeamer ¹⁰

CodeBeamer es una plataforma integral de gestión de proyectos que ofrece una amplia gama de herramientas específicas para equipos de desarrollo de software. Desde la gestión de requisitos hasta el seguimiento de incidencias y la gestión de pruebas, CodeBeamer proporciona una solución completa para planificar, ejecutar y controlar proyectos de desarrollo de software de principio a fin.

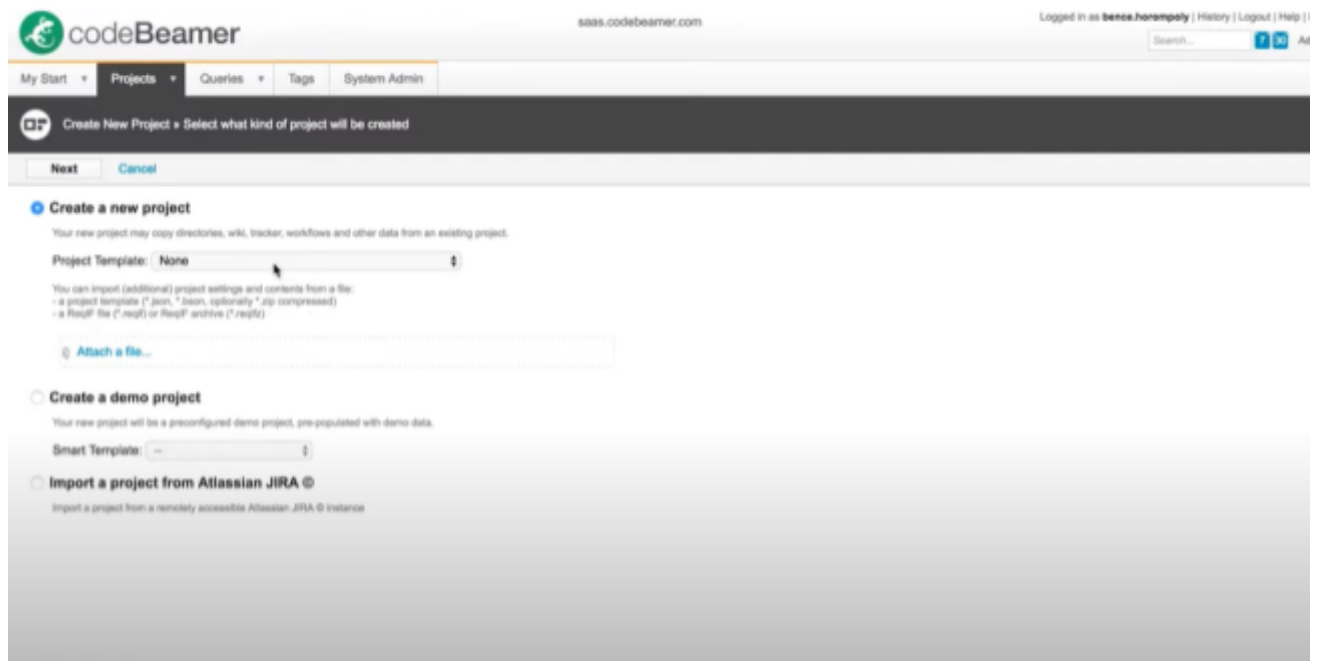


Ilustración 10: Creación de proyectos en codeBeamer

2.2.4.1. Características

Manejo de proyectos

CodeBeamer permite a los equipos de desarrollo crear y gestionar proyectos de manera eficiente, ofreciendo funcionalidades para la creación de estructuras de trabajo, la definición de objetivos y el seguimiento del progreso a través de tableros Kanban y gráficos de Gantt, aunque no brinda la opción de personalizar los proyectos en sí.

Manejo de requisitos

Una de las características destacadas de CodeBeamer es su robusto sistema de gestión de requisitos. Permite a los equipos capturar, organizar y gestionar requisitos de manera estructurada.

Estos requisitos e historias además de poder organizarse en estructuras jerárquicas, los campos para su definición son completamente adaptables

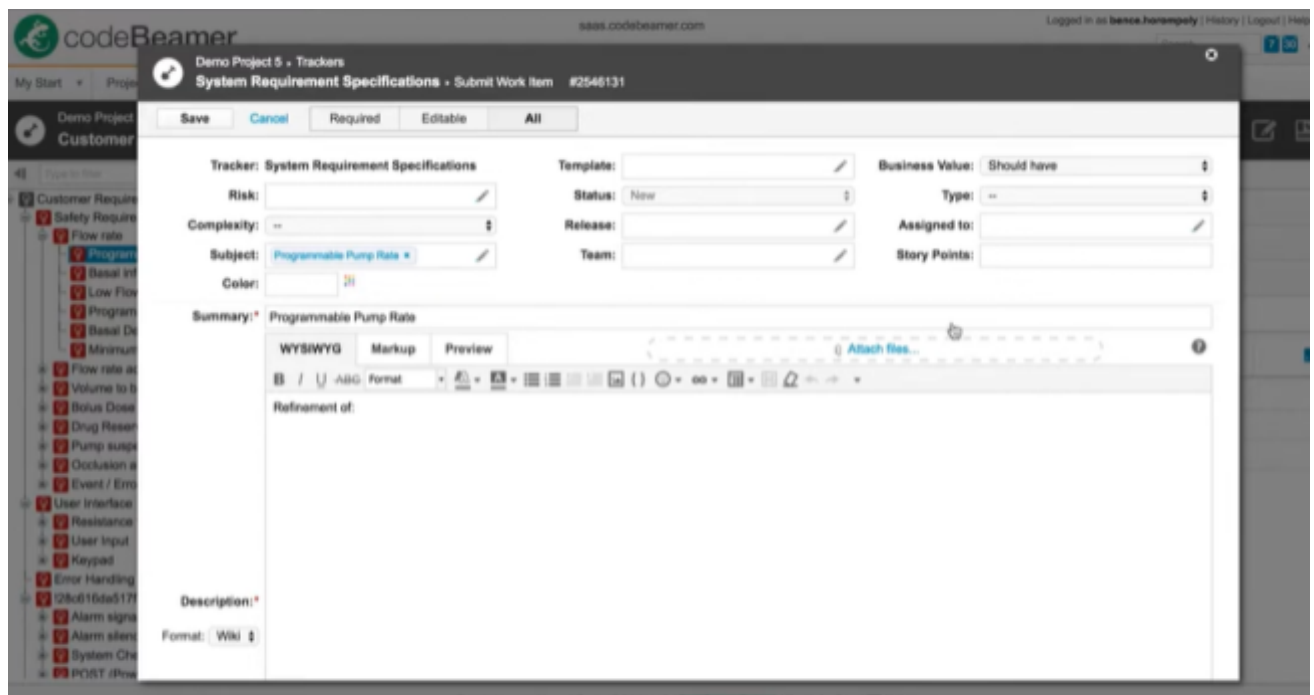


Ilustración 11: Creación de requisitos en codeBeamer

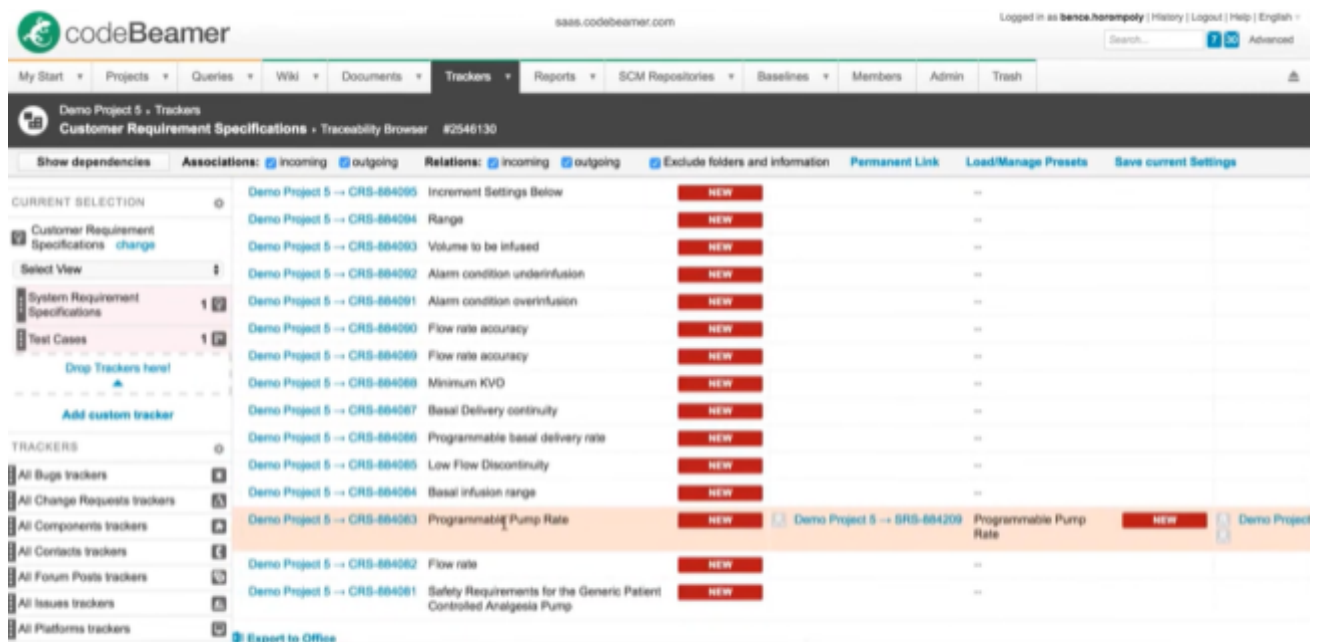


Ilustración 12: Manejo de requisitos en codeBeamer

Seguimiento de incidencias

Ofrece capacidades avanzadas de personalización para adaptar el flujo de trabajo de seguimiento de incidencias a las

necesidades específicas del equipo y del proyecto. Además, cuenta con un motor de búsqueda para localizar las necesidades de manera eficiente

Trazabilidad

CodeBeamer facilita el seguimiento entre necesidades e historial de cambios realizados en estas, y cuenta con un motor de búsqueda para localizar las necesidades de manera eficiente.

Gestión de Usuarios y Roles

CodeBeamer ofrece una robusta plataforma para la gestión de usuarios y roles, lo que permite a los equipos controlar de manera efectiva quién tiene acceso a qué información y qué acciones pueden realizar dentro del sistema. Esta funcionalidad es fundamental para garantizar la seguridad y la colaboración eficiente en proyectos de desarrollo de software y tecnológicos.

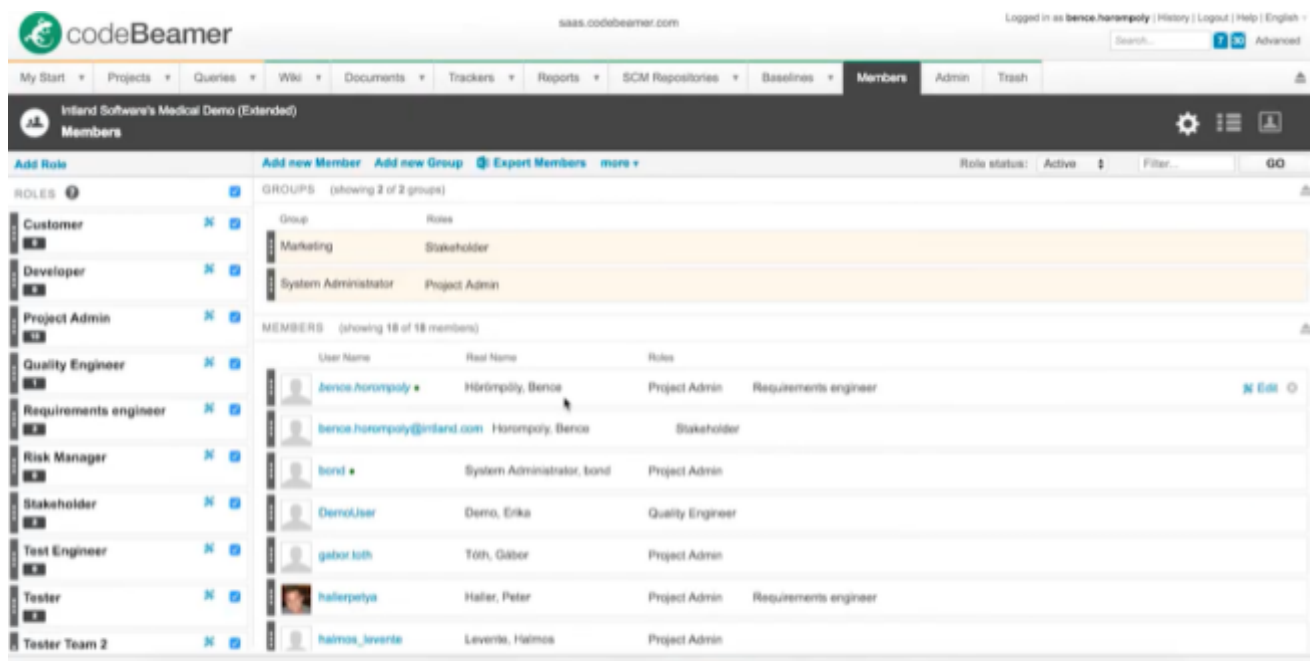


Ilustración 13: Gestión de usuarios en codeBeamer

2.2.4.2. Ventajas

Entre las ventajas de CodeBeamer se incluyen su enfoque específico en el desarrollo de software, su amplia gama de funcionalidades específicas para equipos de desarrollo y su capacidad para adaptarse a diferentes metodologías de desarrollo, como Agile y Waterfall. Además, ofrece una interfaz

intuitiva y fácil de usar, así como una robusta suite de integraciones con otras herramientas de desarrollo.

2.2.4.3. Desventajas

Sin embargo, algunas cosas que pueden ser desventajas de CodeBeamer incluyen que puede ser más caro al principio que otras herramientas similares, que puede tomar un poco de tiempo para aprender a usarlo si eres nuevo, además necesitas ajustarlo y personalizarlo para que funcione bien con tu equipo y tu proyecto.

2.2.4.4. Conclusiones

En resumen, CodeBeamer es una buena opción para equipos que desarrollan software, ya que tiene muchas funciones útiles para manejar proyectos de manera efectiva y asegurarse de que los productos sean de alta calidad. Aunque tiene algunas limitaciones, su enfoque en el desarrollo de software y sus ventajas en términos de funciones específicas hacen que sea una opción popular para equipos que quieren mejorar la eficiencia y calidad de sus proyectos.

3. Planificación del proyecto

En este capítulo realizamos una planificación del proyecto, determinando su alcance, decidiendo la metodología a utilizar, estableciendo un plan temporal y un plan de costes; así mismo, realizaremos la elección de las tecnologías más adecuadas para el desarrollo.

3.1. Alcance del proyecto

El trabajo abarca desde el diseño y desarrollo de la herramienta de trazabilidad de requisitos software hasta la realización de estudios teóricos y experimentales que validen su efectividad, así como la gestión y control eficiente del proyecto para garantizar su correcta ejecución y entrega de resultados satisfactorios.

Por otra parte, la elaboración de una aplicación como esta puede presentar una complejidad considerable para un proyecto académico como el presente, el cual consta de 300 horas de trabajo. Esta circunstancia nos lleva a establecer un alcance definido que posibilite la creación de una aplicación funcional, pero al mismo tiempo, permita su conclusión en un lapso razonable, considerando los recursos limitados del proyecto. Por consiguiente, se ha determinado que este proyecto se centre en el desarrollo de una aplicación elemental que facilite la creación y gestión de proyectos, así como la administración de diversos tipos de requisitos.

Basándonos en los objetivos del producto, detallados en el capítulo 1, las necesidades que se pretenden cubrir y que definen el alcance del proyecto, son las siguientes:

- Desarrollo de las funcionalidades principales del sistema, como la pantalla de inicio de sesión y registro, menú desplegable, y sistema de cierre de sesión.
- Verificación de la compatibilidad con diferentes navegadores y dispositivos.
- Desarrollo de las opciones de creación de proyectos, grupos y usuarios.
- Desarrollo de la herramienta de creación de épicas (agrupaciones de requisitos), requisitos y su trazabilidad.
- Optimización del rendimiento y la seguridad del sistema.

3.2. Metodología utilizada.

Para mi proyecto de fin de grado, he optado por utilizar la metodología de desarrollo iterativo incremental ¹¹. Esta elección se basa en la necesidad de abordar el proyecto de manera flexible y adaptable.

La metodología iterativa incremental implica dividir el proyecto en partes más pequeñas, llamadas iteraciones. Cada iteración produce un incremento en la funcionalidad del software. Esto significa que puedo trabajar en el proyecto paso a paso, abordando y completando partes específicas en cada fase.

Una de las razones por las que elegí esta metodología es su capacidad para adaptarse a cambios en los requisitos del proyecto. Siempre que surja nueva información o necesidades adicionales, puedo ajustar el proyecto durante las iteraciones posteriores. Esto me da la flexibilidad necesaria para responder a cualquier imprevisto en los requisitos de manera oportuna y eficiente.

Además, al dividir el proyecto en partes más pequeñas, puedo obtener retroalimentación temprana y regular de mi supervisor y otros interesados. Esto me ayuda a asegurarme de que el proyecto esté en el camino correcto y que cumpla con las expectativas del cliente o usuario final.

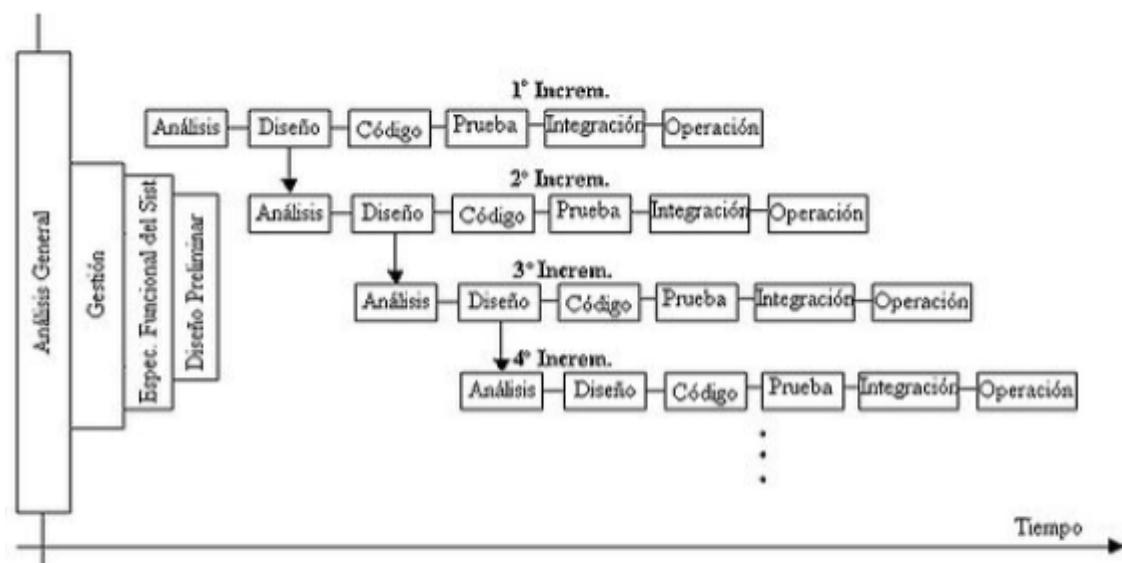


Ilustración 14: Desarrollo iterativo incremental

En resumen, la metodología de desarrollo iterativo incremental me permite abordar mi proyecto de manera flexible, adaptativa y enfocada en la entrega gradual de funcionalidades. Esto garantiza que pueda cumplir con los requisitos del proyecto de manera efectiva y lograr los objetivos establecidos.

3.3. Planificación temporal

Para la planificación de este proyecto, se ha adoptado un enfoque basado en la identificación y segmentación de tareas clave en iteraciones sucesivas. Este enfoque permite una gestión más efectiva del tiempo y los recursos, así como la adaptación continua a medida que avanza el desarrollo del proyecto.

Iteración 1: Fundamentos y Diseño Inicial

➤ **Ingeniería de Requisitos:**

- Definición de la estructura y objetivos del proyecto.
- Identificación y análisis de los requisitos iniciales del sistema.

➤ **Diseño e Implementación:**

- Desarrollo de los primeros prototipos y maquetas.
- Diseño preliminar de la arquitectura del sistema.
- Diseño del diagrama de clases y del modelo relacional de la base de datos.
- Selección y configuración de las herramientas y tecnologías a utilizar.
- Desarrollo de las funcionalidades principales del sistema, como la pantalla de inicio de sesión y registro, menú desplegable, y sistema de cierre de sesión.

➤ **Pruebas:**

- Realización de pruebas de funcionalidad básica de los prototipos.
- Verificación de la compatibilidad con diferentes navegadores y dispositivos.

➤ **Documentación:**

- Diseño inicial del documento de especificaciones del proyecto.
- Registro de las decisiones de diseño y arquitectura.

Iteración 2: Desarrollo y Refinamiento

➤ **Ingeniería de Requisitos:**

- Revisión y refinamiento de los requisitos del sistema.
- Incorporación de nuevos requisitos surgidos del análisis preliminar.

➤ **Diseño e Implementación:**

- Desarrollo de las opciones de creación de proyectos, grupos y usuarios.
- Integración de componentes y módulos desarrollados en la iteración anterior.
- Modificación del diagrama de clases de acuerdo a las funcionalidades incorporadas.
- Mejoras en la usabilidad y experiencia del usuario en la aplicación.

➤ **Pruebas:**

- Pruebas exhaustivas de todas las funcionalidades implementadas.
- Identificación y corrección de errores y fallos.

➤ **Documentación:**

- Ampliación de la documentación de especificaciones del proyecto.
- Descripción detallada de las funcionalidades implementadas.

Iteración 3: Finalización y Entrega

➤ **Ingeniería de Requisitos:**

- Validación final de los requisitos del sistema.
- Ajustes finales basados en la retroalimentación recibida.
- Actualización del análisis definido en la segunda iteración para adecuarlo a la tercera iteración.

➤ **Diseño e Implementación:**

- Desarrollo de la herramienta de creación de epicas, requisitos y su trazabilidad.
- Optimización del rendimiento y la seguridad del sistema.
- Implementación de características adicionales según la retroalimentación del usuario.
- Modificación del diagrama de clases.
- Implementación de las clases definidas en esta iteración.

➤ **Pruebas:**

- Pruebas de aceptación del cliente.
- Garantía de calidad final y solución de problemas pendientes.

➤ **Documentación:**

- Preparación de la documentación de usuario final.

- Finalización del documento de especificaciones del proyecto.

3.4. Tecnologías utilizadas

El entorno tecnológico de un proyecto es fundamental para comprender su arquitectura y el conjunto de herramientas utilizadas en su desarrollo. En el caso de mi Trabajo de Fin de Grado (TFG), he optado por un entorno versátil y potente que incluye diversas tecnologías líderes en el desarrollo web.

Para la construcción del TFG, he empleado Visual Studio Code como mi entorno de desarrollo integrado (IDE, por sus siglas en inglés). Visual Studio Code ofrece una amplia gama de características y extensiones que facilitan la escritura de código en múltiples lenguajes, lo que lo convierte en una elección popular entre los desarrolladores.

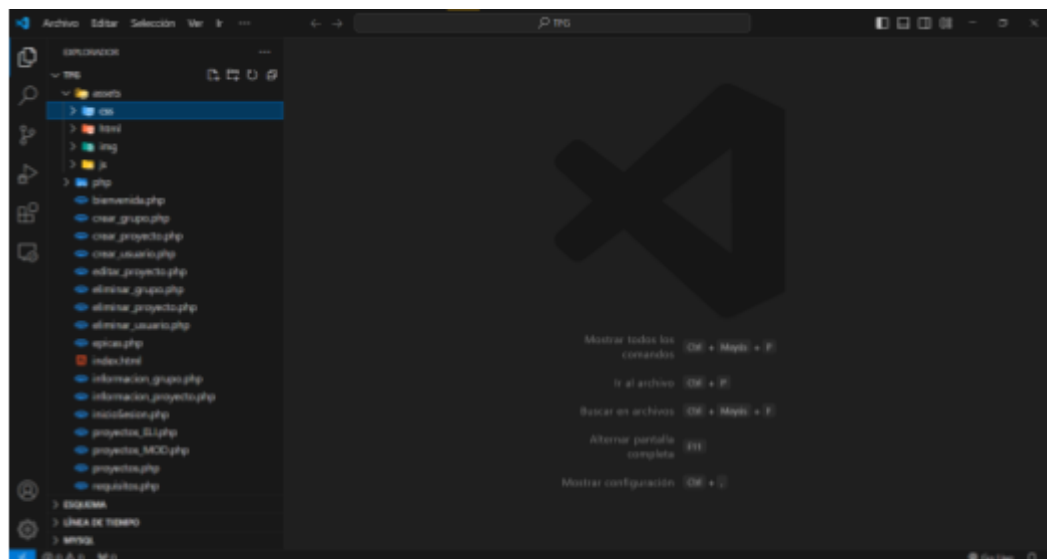


Ilustración 15: Visual Studio Code

En cuanto a los lenguajes de programación, he utilizado PHP para la lógica del lado del servidor, CSS para el diseño y la presentación de la interfaz de usuario, y JavaScript para la interactividad y la mejora de la experiencia del usuario. Estos lenguajes forman la base de la aplicación web, permitiendo la creación de un sistema dinámico y funcional.

Para gestionar los datos de la aplicación, he empleado una base de datos MySQL, una opción sólida y ampliamente utilizada en la comunidad de desarrollo web. MySQL proporciona un entorno robusto para el almacenamiento y la manipulación de datos, lo que resulta fundamental para cualquier aplicación que requiera persistencia de información.



Ilustración 16: Lenguaje de programación

Para la implementación y pruebas locales de la aplicación, he utilizado el servidor XAMPP, que incluye Apache como servidor web, MySQL como sistema de gestión de bases de datos y PHP como lenguaje de script del lado del servidor. XAMPP proporciona un entorno de desarrollo completo y fácil de configurar, lo que facilita la creación y prueba de la aplicación en mi propio equipo antes de su despliegue en un entorno de producción.

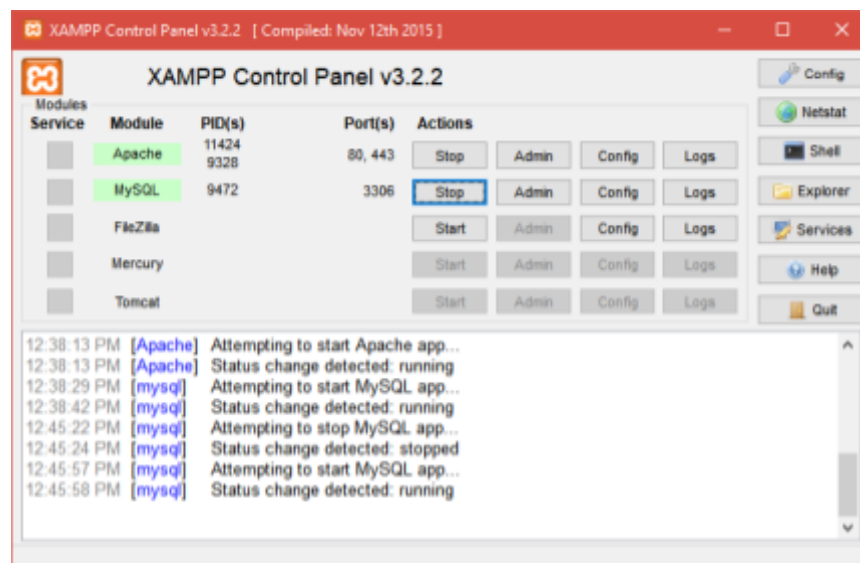


Ilustración 17: Pantalla XAMPP

En resumen, el entorno tecnológico de mi TFG se compone de Visual Studio Code como IDE, PHP, CSS y JavaScript como lenguajes de programación, MySQL como sistema de gestión de bases de datos, y XAMPP como servidor local. Esta combinación de herramientas proporciona una plataforma sólida y eficiente para el desarrollo de aplicaciones web modernas y funcionales.

3.5. Valoración de la carga de trabajo requerida

En la planificación temporal se ha realizado un desglose de las tareas del proyecto. Nos proponemos ahora realizar una estimación de la carga de trabajo requerida para desempeñar estas tareas, utilizando la técnica de **puntos de función**.¹²

Los puntos de función son una técnica de estimación que se basa en la funcionalidad proporcionada por un sistema de software. Se evalúan diferentes aspectos de la funcionalidad, como el número de entradas y salidas del sistema, la cantidad de datos procesados y la complejidad de las consultas. Estos aspectos se ponderan utilizando factores de ajuste que reflejan características específicas del proyecto, como la complejidad de los datos o la experiencia del equipo de desarrollo.

En este caso, se menciona que se utilizará un factor de ajuste constante para todas las iteraciones del proyecto. Este factor de ajuste se compone de 14 factores diferentes, cada uno con un valor asignado específico. Estos factores se utilizan para ajustar la estimación de puntos de función y tener en cuenta las características únicas del proyecto que pueden influir en la cantidad de esfuerzo necesario para completarlo. Cada uno de estos factores se evalúa en una escala de valores predefinidos que van desde "Bajo" hasta "Alto", o de "1" a "5" por ejemplo, dependiendo de la metodología específica de puntos de función que se esté utilizando.

FACTOR	FC
1. Comunicación de datos	2
2. Procesamiento distribuido	0
3. Eficiencia en el procesamiento	4
4. Entrada de datos en línea	5
5. Actualización en línea	4
6. Complejidad de consulta	2
7. Informes complejos	3
8. Facilidad de cambio	2
9. Consultas ad hoc	5
10. Cambios de procesamiento	3
11. Reutilización de código	0

12. Facilidad de instalación	5
13. Facilidad de operación	3
14. Portabilidad	3

Factor de Ajuste de Valor o FA ($0.01 * \Sigma FC$) = 0,41

Donde:

- El **0.01** en la fórmula del Factor de Ajuste de Valor (FA) proviene del modelo de cálculo estándar para los puntos de función.

El cálculo estimado para cada ciclo es el siguiente:

***Nota:** las estimaciones del grado de complejidad las he establecido siendo asesorado por el tutor del TFG y por mi mismo al tener experiencia previa en este tipo de desarrollos.

Iteración 1

- **Entradas:**

ENTRADA	COMPLEJIDAD
Programación pantalla inicial	Media
Elaboración del footer	Media
Elaboración de ejemplos de trabajos	Baja
Creación de botón inicio sesión	Media
Programación página registro de admin	Media
Programación inicio de sesión	Alta
Cambio de Inicio a Registro	Alta
Creación pantalla principal de la web	Media
Programación caja con usuario registrado	Media
Programación cierre de sesión	Alta
Creación menu lateral	Alta

- **Salidas:**

SALIDA	COMPLEJIDAD
Visualización pantalla inicial	Media
Visualización del footer	Media
Visualización inicio sesión y registro	Baja
Visualización pantalla principal de la web	Media
Visualización menú lateral	Alta

- **Consultas:**

CONSULTA	COMPLEJIDAD
Interfaz botón de inicio	Baja
Interfaz inicio sesión y registro	Alta
Interfaz del menú lateral	Media
Interfaz botón cierre de sesión	Alta

- **Ficheros lógicos internos:**

CONSULTA	COMPLEJIDAD
Base de datos	Alta

Con la información obtenida, podemos calcular los puntos de función sin ajustar según los tipos de elementos (Entradas, Salidas, Consultas, Ficheros Lógicos Internos) y sus niveles de complejidad (Baja, Media, Alta). Los números dentro de la tabla corresponden a la cantidad de cada tipo de elemento, el factor de peso según la complejidad, y el total que resulta de multiplicar la cantidad por el factor de peso.

ELEMENTO	COMPLEJIDAD	NÚMERO	TOTAL
Entradas	Baja(x3)	1	3
	Media(x4)	6	24
	Alta(x6)	4	24
Salidas	Baja(x4)	1	4
	Media(x5)	3	15
	Alta(x7)	1	7
Consultas	Baja(x3)	1	3
	Media(x4)	1	4
	Alta(x6)	2	12
Ficheros lógicos internos	Baja(x7)	0	0
	Media(x10)	0	0
	Alta(x15)	1	15

Puntos de función sin ajustar (PFSA)= 111

Considerando los datos de esta tabla, y tomando en cuenta los criterios previamente mencionados, se derivan los siguientes cálculos:

- Puntos de función ajustados:**

Si aplicamos el factor de corrección a los puntos de función sin ajustar obtenemos el siguiente resultado:

$$PFSA * (FA + 0.65) = 111 * (0.41 + 0.65) = 117.66 PFA$$

Donde:

- **PFSA** (Puntos de Función sin Ajustar): Es el total de puntos de función calculados antes de aplicar cualquier factor de ajuste. En tu caso, es 111.
- **FA** (Factor de Ajuste): Se calcula sumando los valores asignados a 14 características de complejidad general del sistema.
- El valor de **0.65** es parte de una fórmula estándar utilizada en la estimación de puntos de función ajustados. Esta fórmula se deriva

del modelo de análisis de puntos de función de Albrecht, que toma en cuenta factores de ajuste de complejidad general del sistema. La constante 0.65 se usa para asegurar que el ajuste no sea negativo, garantizando que siempre se incrementen o se mantengan los puntos de función después de considerar los factores de ajuste.

- **Esfuerzo en horas/persona:**

Después de obtener los puntos de función ajustados, podemos calcular la estimación de las horas necesarias para llevar a cabo esta iteración de la siguiente manera:

$$PFSA * \left(\frac{2.5 \text{ horas}}{1 \text{ persona}} \right) = 117.66 * (2.5) = 294.15 \text{ horas}$$

Donde:

- El valor de **2,5 horas** por punto de función es una media basada en datos históricos y estudios de productividad en proyectos de software que utilizan lenguajes de 4ª generación. Este valor puede ser ajustado según las características particulares de cada proyecto, la experiencia del equipo de desarrollo y las herramientas utilizadas.

Iteración 2

- **Entradas:**

ENTRADA	COMPLEJIDAD
Crear un proyecto	Alta
Visualización de todos los proyectos	Media
Modificación de un proyecto	Media
Eliminación de un proyecto	Media
Añadir botones al menú para lo anterior	Baja
Botón para visualizar el proyecto	Baja
Creación de los grupos de trabajo	Alta
Creación listado de grupos de trabajo	Media
Asignación de un proyecto a un grupo	Media
Creación de un nuevo usuario	Alta

Asignación de rol a usuario	Media
Asignación de usuario a grupo de trabajo	Media
Opción eliminación de usuario	Media
Opción eliminación de grupo de trabajo	Media
Añadir botones al menú para lo anterior	Baja

- **Salidas:**

SALIDA	COMPLEJIDAD
Visualización de proyectos	Media
Visualización de usuarios	Media
Visualización grupos de trabajo	Media
Visualización nuevo menú lateral	Baja

- **Consultas:**

CONSULTA	COMPLEJIDAD
Interfaz creación de Proyectos	Media
Interfaz creación de Grupos de trabajo	Baja
Interfaz creación de Usuario	Media
Interfaz de listado de Proyectos	Baja
Interfaz de listado de Grupos	Baja
Interfaz de listado de Usuarios	Baja

- **Ficheros lógicos internos:**

CONSULTA	COMPLEJIDAD
Base de datos	Alta

Con la información obtenida, podemos calcular los puntos de función sin ajustar:

ELEMENTO	COMPLEJIDAD	NÚMERO	TOTAL
Entradas	Baja(x3)	3	9
	Media(x4)	9	36
	Alta(x6)	3	18
Salidas	Baja(x4)	1	4
	Media(x5)	3	15
	Alta(x7)	0	0
Consultas	Baja(x3)	4	12
	Media(x4)	2	8
	Alta(x6)	0	0
Ficheros lógicos internos	Baja(x7)	0	0
	Media(x10)	0	0
	Alta(x15)	1	15

Puntos de función sin ajustar (PFSA)= 117

Considerando los datos de esta tabla, y tomando en cuenta los criterios previamente mencionados, se derivan los siguientes cálculos:

- Puntos de función ajustados:**

Si aplicamos el factor de corrección a los puntos de función sin ajustar obtenemos el siguiente resultado:

$$PFSA * (FA + 0.65) = 117 * (0.41 + 0.65) = 124.02 PFA$$

- Esfuerzo en horas/persona:**

Después de obtener los puntos de función ajustados, podemos calcular la estimación de las horas necesarias para llevar a cabo esta iteración de la siguiente manera:

$$PFSA * \left(\frac{2.5 \text{ horas}}{1 \text{ persona}} \right) = 124.02 * (2.5) = 310.05 \text{ horas}$$

Iteración 3

- **Entradas:**

ENTRADA	COMPLEJIDAD
Pantalla principal del Proyecto	Media
Apartados para Épicas y Requisitos	Baja
Creación de nueva Épica	Media
Listado de Épicas existentes	Media
Creación de nuevo Requisito	Alta
Listado de Requisitos	Media
Pantalla principal del Requisito	Media
Programación de la Trazabilidad	Alta
Creación de los Artefactos del Requisito	Media
Actualización de Trazabilidad con los artefactos creados	Media
Edición del estado del Requisito	Media
Actualización menú lateral	Media

- **Salidas:**

SALIDA	COMPLEJIDAD
Visualización del Proyecto	Media
Visualización Épicas	Media
Visualización listado Requisitos	Media
Visualización trazabilidad	Media
Visualización añadir artefactos	Alta
Visualización trazabilidad final	Alta

- **Consultas:**

CONSULTA	COMPLEJIDAD
Interfaz creación Épica	Baja
Interfaz creación Requisitos	Alta
Interfaz estado del Requisito	Media
Interfaz Trazabilidad	Alta

- **Ficheros lógicos internos:**

CONSULTA	COMPLEJIDAD
Base de datos	Alta

Con la información obtenida, podemos calcular los puntos de función sin ajustar:

ELEMENTO	COMPLEJIDAD	NÚMERO	TOTAL
Entradas	Baja(x3)	1	3
	Media(x4)	9	36
	Alta(x6)	2	12
Salidas	Baja(x4)	0	0
	Media(x5)	4	20
	Alta(x7)	2	14
Consultas	Baja(x3)	1	3
	Media(x4)	1	4
	Alta(x6)	2	12
Ficheros lógicos internos	Baja(x7)	0	0
	Media(x10)	0	0
	Alta(x15)	1	15

Puntos de función sin ajustar (PFSA)= 119

Considerando los datos de esta tabla, y tomando en cuenta los criterios previamente mencionados, se derivan los siguientes cálculos:

- **Puntos de función ajustados:**

Si aplicamos el factor de corrección a los puntos de función sin ajustar obtenemos el siguiente resultado:

$$PFSA * (FA + 0.65) = 119 * (0.41 + 0.65) = 126.14 PFA$$

- **Esfuerzo en horas/persona:**

Después de obtener los puntos de función ajustados, podemos calcular la estimación de las horas necesarias para llevar a cabo esta iteración de la siguiente manera:

$$PFSA * \left(\frac{2.5 \text{ horas}}{1 \text{ persona}}\right) = 126.14 * (2.5) = 315.35 \text{ horas}$$

Con todos estos datos podemos calcular el tiempo estimado total dedicado al proyecto:

- Tiempo estimado necesario para la primera iteración: **294.15 Horas.**
- Tiempo estimado necesario para la segunda iteración: **310.05 Horas.**
- Tiempo estimado necesario para la tercera iteración: **315.35 Horas.**
- Tiempo estimado TOTAL:

$$\Sigma \text{ Horas por iteración} = 294.15 + 310.05 + 315.35 = 919.55 \text{ HORAS}$$

Resumiendo la planificación, tras realizar un cálculo de las horas estimadas, se llega a la conclusión de que el proyecto requerirá un total aproximado de 919.55 horas para su completa implementación y desarrollo.

Con esto podemos estimar que, suponiendo **4 horas de trabajo por día**, nos sale unos **230 días de trabajo o unos 7 meses y medio**, desde que comenzó en octubre.

3.6. Presupuesto

En este apartado, explicaremos cómo determinar cuánto dinero se necesita para llevar a cabo cada etapa del proyecto, desde el principio hasta el final.

El objetivo es entender cómo planificar y asignar el dinero necesario para garantizar que el proyecto se complete de manera exitosa y dentro del presupuesto establecido y dividido en 2 bloques: hardware y software.

3.6.1. Coste hardware

Los elementos que se van a utilizar en este proyecto son los siguientes:

- Ordenador personal: HP Pavilion Power Laptop 15-cb0xx, tamaño pantalla (pulgadas) : 15, almacenamiento : 1256 GB, velocidad del procesador : 2.8 GHz, procesador Intel i7, sistema operativo : Windows.
- Conexión a internet: conexión de fibra óptica de 600Mbps, pondremos solo un 20% del total gastado para no excedernos.
- Gastos de impresión en copistería, sobre unos 20€ ya que casi todo el formato es online.
- Luz. Se estima que el uso aproximado del 20% también al no utilizar toda la capacidad durante el proceso.

A continuación, se muestra una tabla con el coste estimado de cada recurso y del coste total de los componentes hardware.

Hardware	Uso (%)	Coste Total(€)	Coste(€)
Ordenador personal	15%	1200€	180€
Conexión a internet	20%	360€	72€
Gastos impresión	-	20€	20€
Gastos Luz	20%	620€	124€

3.6.2. Coste software

El software utilizado es el siguiente:

- MySQL.
- Adobe Photoshop.
- Adobe Illustrator.
- Sistema operativo Windows 11.
- Visual Paradigm.
- Visual Studio Code
- XAMP
- Navegadores web.
- Docker.
- Adobe Acrobat Pro.
- Microsoft Office 365.

A continuación, se muestra una tabla con el coste estimado de cada recurso y del coste total de los componentes software.

Hardware	Uso (%)	Coste Total(€)	Coste(€)
MySQL	80%	0€	0€
Adobe Photoshop	10%	0€	0€
Adobe Illustrator	60%	0€	0€
Sistema operativo Windows 11	40%	259€	103.6€
Visual Paradigm	10%	0€	0€
Visual Studio Code	90%	0€	0€
XAMP	90%	0€	0€
Navegadores web	90%	0€	0€
Docker	20%	0€	0€
Adobe Acrobat Pro	80%	0€	0€
Microsoft Office 365	10%	69€	6.3€

3.6.3. Coste total estimado

A continuación, se muestra una tabla con el coste estimado total del proyecto.

	Coste(€)
Hardware	396€
Software	109.9€
TOTAL	505.9€

4. ANÁLISIS

En este capítulo se describe el proceso de análisis que se ha llevado a cabo en las sucesivas iteraciones en las que se ha desarrollado el proyecto.

En la fase de análisis, la principal tarea a llevar a cabo es la obtención de los requisitos, sobre todo funcionales, que debe cumplir la aplicación a desarrollar. En este caso, los requisitos dictan la norma sobre lo que debe ejecutar el producto a crear.

4.1. Épicas y requisitos

4.1.1. Épicas

Para desarrollar una aplicación de gestión de requisitos que cumpla con las expectativas y necesidades del cliente, se deben cubrir las siguientes áreas clave que son conocidas como épicas. Cada épica será una agrupación de requisitos o casos de uso:

- **EPIC-01 Registro, EPIC-02 Autenticación:** es fundamental implementar un sistema robusto que permita el registro y autenticación de usuarios. Esto garantizará que solo los usuarios autorizados puedan realizar acciones sensibles, como la eliminación de usuarios y proyectos.
- **EPIC-03 Gestión de usuarios:** debe permitir la creación, edición y eliminación de usuarios de manera eficiente. Además, es esencial contar con una funcionalidad que permita asignar usuarios a grupos específicos, facilitando así la organización y administración dentro de la aplicación.
- **EPIC-04 Gestión de grupos:** la aplicación debe ofrecer la capacidad de crear, editar y eliminar grupos. Es importante también permitir la asignación de usuarios a grupos y de grupos a proyectos, asegurando una organización adecuada y facilitando la gestión de permisos y roles dentro de cada proyecto.
- **EPIC-05 Gestión de proyectos:** la aplicación debe permitir la creación, edición y eliminación de proyectos con detalles completos como nombre, descripción y fechas relevantes. Además, debe ser posible asignar grupos a proyectos, lo que permitirá una mejor organización y seguimiento del trabajo en equipo.
- **EPIC-06 Gestión de épicas:** debe incluir la capacidad de crear y eliminar épicas dentro de los proyectos. Esta funcionalidad es crucial para organizar y desglosar el trabajo en tareas más manejables, facilitando la planificación y ejecución de los proyectos.

- **EPIC-07 Gestión de requisitos:** la aplicación debe permitir la creación, edición y eliminación de requisitos dentro de las épicas. Además, debe incluir la capacidad de añadir y gestionar detalles específicos de los requisitos, llamados **artefactos**, como casos de uso, prototipos de interfaz de usuario, diagramas de clases y casos de prueba. También es esencial proporcionar una opción para ver el estado actual de los requisitos y comprobar su trazabilidad a lo largo del ciclo de vida del proyecto.

En resumen, la aplicación debe ser segura, fácil de usar y permitir una gestión integral de usuarios, grupos, proyectos, épicas y requisitos. Todo esto, con el fin de conseguir una buena trazabilidad para el seguimiento del progreso del proyecto, ofreciendo una interfaz intuitiva, asegurando una gestión eficiente de los requisitos de los proyectos de software.

4.1.2. Requisitos

Vamos a descomponer las épicas en los requisitos funcionales específicos que son requeridos para satisfacer el objetivo al que se refiere cada épica.

FRQ-01: Registro en el sistema: es necesario registrarse en el sistema para poder iniciar sesión.

FRQ-02: Inicio de sesión: es necesario iniciar sesión en el sistema para acceder a las funcionalidades de este.

FRQ-03: Cierre de sesión: para poder salir del sistema una vez finalizado el trabajo.

FRQ-04: Crear nuevo usuario: incluyendo como atributos: identificador numérico, nombre, rol, teléfono, email, foto. El usuario recién creado deberá asociarse a un grupo de trabajo ya registrado.

FRQ-05: Eliminar usuario: permitirá borrar un usuario registrado del sistema.

FRQ-06: Asignar rol a usuario: Los posibles roles son: usuario regular y jefe de proyecto.

FRQ-07: Visualizar usuarios existentes: El sistema presentará una tabla con los usuarios registrados.

FRQ-08: Asignar grupo de trabajo a usuario: Los usuarios se agruparán en “grupos de trabajo”. Se dispondrá de la función para asignar grupo a un usuario ya registrado.

FRQ-09: Crear nuevo grupo de trabajo: Identificado por un id, teniendo como atributos: nombre y foto, y estará asociado a un proyecto ya creado. Cada grupo de trabajo tendrá un solo jefe de proyecto.

FRQ-10: Eliminar grupo de trabajo: permitirá eliminar un grupo que no tenga ningún usuario asociado.

FRQ-11: Visualizar grupos de trabajo existentes: El sistema mostrará una tabla con todos los grupos registrados en el sistema.

FRQ-12: Ver información del grupo de trabajo: Se mostrará el grupo de trabajo con todos sus componentes.

FRQ-13: Asignar proyecto a un grupo de trabajo: teniendo en cuenta que cada grupo tendrá un solo proyecto y cada proyecto corresponderá a un solo grupo.

FRQ-14: Crear nuevo proyecto: que tendrá un identificador numérico, una descripción y la metodología utilizada. El jefe de proyecto será el propio usuario que lo haya creado.

FRQ-15: Modificar proyecto: permitirá modificar un proyecto registrado del sistema.

FRQ-16: Eliminar proyecto: permitirá borrar un proyecto registrado del sistema.

FRQ-17: Visualizar proyectos existentes: El sistema mostrará una tabla con todos los proyectos registrados en el sistema.

FRQ-18: Ver información del proyecto: Se mostrará el proyecto con todos sus componentes.

FRQ-19: Crear nueva épica: que estará identificada por un número, tendrá un nombre y una descripción y corresponderá a un proyecto.

FRQ-20: Visualizar lista de épicas: El sistema mostrará una tabla con todas las épicas registradas en el sistema.

FRQ-21: Ver requisitos asociados a las épicas

FRQ-22: Crear nuevo requisito: que se corresponderá con una determinada épica. Estará identificado por un número y tendrá un nombre y una descripción. Tendrá cuatro estados posibles: propuesto, en revisión, en desarrollo y finalizado.

FRQ-23: Visualizar listado de requisitos: El sistema mostrará una tabla con todos los requisitos registrados en el sistema.

FRQ-24: Ver información del requisito: Se mostrará el requisito con todos sus componentes.

FRQ-25: Cambiar el estado del requisito: Se podrá cambiar el estado del requisito cuando se desee.

FRQ-26: Visualizar la trazabilidad del requisito: Se mostrará la trazabilidad del requisito con todos sus artefactos.

FRQ-27: Añadir caso de uso al requisito: Se podrá añadir tantos casos de uso como se quiera al requisito.

FRQ-28: Añadir diagrama de clase al requisito: Se podrá añadir tantos diagramas de clases como se quiera al requisito.

FRQ-29: Añadir prototipo de interfaz de usuario al requisito: Se podrá añadir tantos prototipos de interfaz de usuario como se quiera al requisito.

FRQ-30: Añadir caso de prueba al requisito: Se podrá añadir tantos casos de prueba como se quiera al requisito.

4.2. Casos de uso.

A través de los casos de uso, se identifican las funciones principales del sistema y se establecen los límites de su comportamiento. Cada caso de uso está asociado a un requisito (FRQ-X) y forma parte de una determinada épica (EPIC-X).

Casos de uso:

- **EPIC-01 Registro**
 - FRQ-01: Registro en el sistema
- **EPIC-02 Autenticación.**
 - FRQ-02: Inicio de sesión
 - FRQ-03: Cierre de sesión
- **EPIC-03 Gestión de usuarios:**
 - FRQ-04: Crear nuevo usuario
 - FRQ-05: Eliminar usuario
 - FRQ-06: Asignar rol a usuario
 - FRQ-07: Visualizar usuarios existentes
 - FRQ-08: Asignar grupo de trabajo a usuario
- **EPIC-04 Gestión de grupos:**
 - FRQ-09: Crear nuevo grupo de trabajo
 - FRQ-10: Eliminar grupo de trabajo
 - FRQ-11: Visualizar grupos de trabajo existentes
 - FRQ-12: Ver información del grupo de trabajo
 - FRQ-13: Asignar proyecto a un grupo de trabajo
- **EPIC-05 Gestión de proyectos:**
 - FRQ-14: Crear nuevo proyecto
 - FRQ-15: Modificar proyecto
 - FRQ-16: Eliminar proyecto
 - FRQ-17: Visualizar proyectos existentes
 - FRQ-18: Ver información del proyecto
- **EPIC-06 Gestión de épicas:**
 - FRQ-19: Crear nueva épica
 - FRQ-20: Visualizar lista de epicas
 - FRQ-21: Ver requisitos asociados a las épicas
- **EPIC-07 Gestión de requisitos:**
 - FRQ-22: Crear nuevo requisito
 - FRQ-23: Visualizar listado de requisitos

- FRQ-24: Ver informacion del requisito
- FRQ-25: Cambiar el estado del requisito
- FRQ-26: Visualizar la trazabilidad del requisito
- FRQ-27: Añadir caso de uso al requisito
- FRQ-28: Añadir diagrama de clase al requisito
- FRQ-29: Añadir prototipo de interfaz de usuario al requisito
- FRQ-30: Añadir caso de prueba al requisito

4.3. Diagrama de casos de uso.

Este diagrama proporciona una visión general de los diferentes roles que interactúan con el sistema y los diversos escenarios de uso que se pueden encontrar.

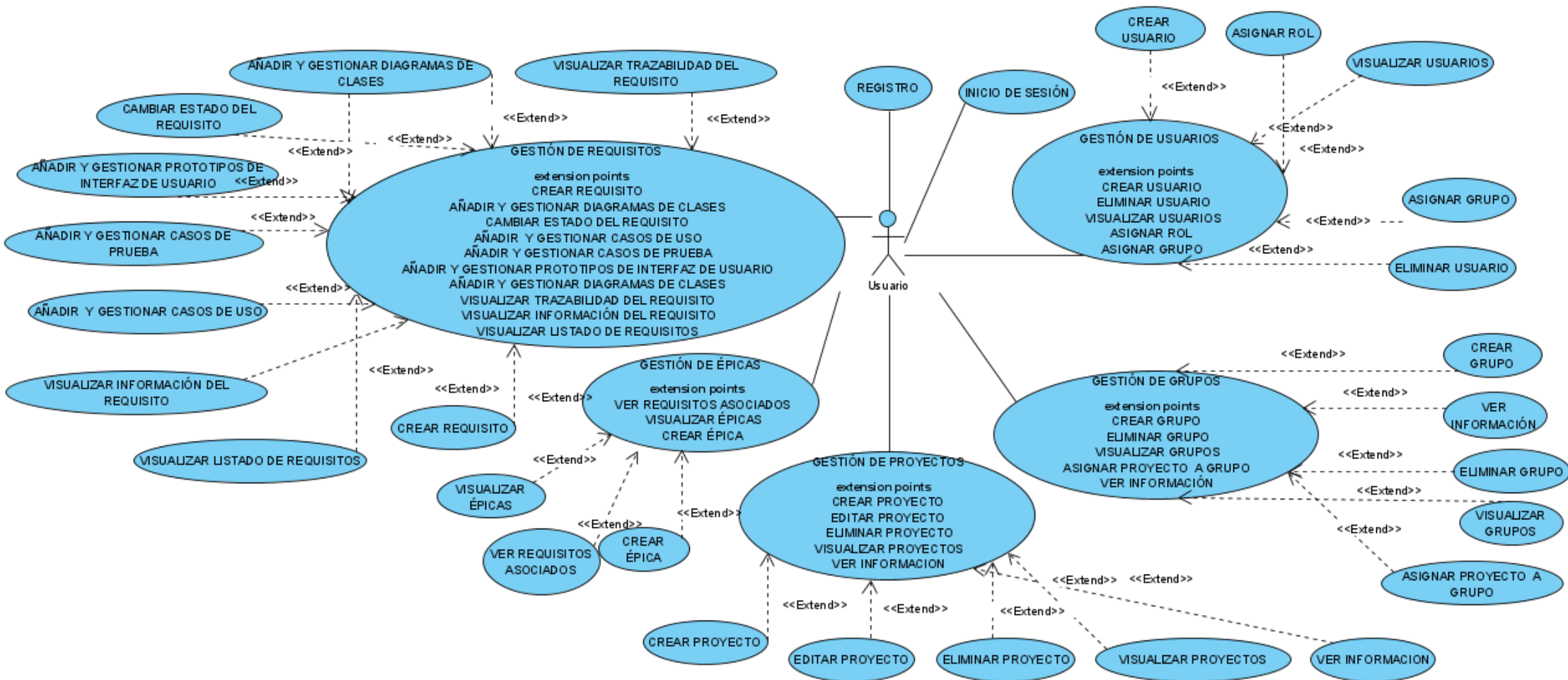


Ilustración 18: Diagrama de casos de uso

4.4. Actores del sistema

Aquí describiremos los actores implicados en el uso del sistema:

- **Usuario no identificado:** El actor Usuario representa a cualquier potencial usuario del sistema que acceda a éste sin necesidad de estar registrado. Su función dentro del sistema es la capacidad de identificarse.
- **Usuario identificado:** El actor Usuario identificado representa a los usuarios que están registrados en el sistema, y que además han seguido un proceso de identificación.
- **Jefe del proyecto:** Este, será la persona que cree un nuevo proyecto, automáticamente se le asociará a él y podrá manejar y realizar todas las operaciones que ofrece la aplicación.



Ilustración 19: Actores del sistema

4.5. Modelado Entidad/Relación

En esta sección se describe de manera detallada todo el proceso de análisis relacionado con los datos que se utilizarán en esta aplicación.

Comenzaremos con un diagrama Entidad-Relación (ER) que ilustra la estructura de los datos necesarios para el proyecto. Posteriormente, se desarrollará este diagrama, estableciendo los requisitos de información y determinando los atributos que definirán la información que procesará la aplicación.

4.5.1. Diagrama entidad/relación

El diagrama ER es una representación visual de las entidades, atributos y relaciones entre ellas dentro del sistema de datos. Este diagrama ayuda a comprender la estructura y la interconexión de los datos esenciales para la aplicación.

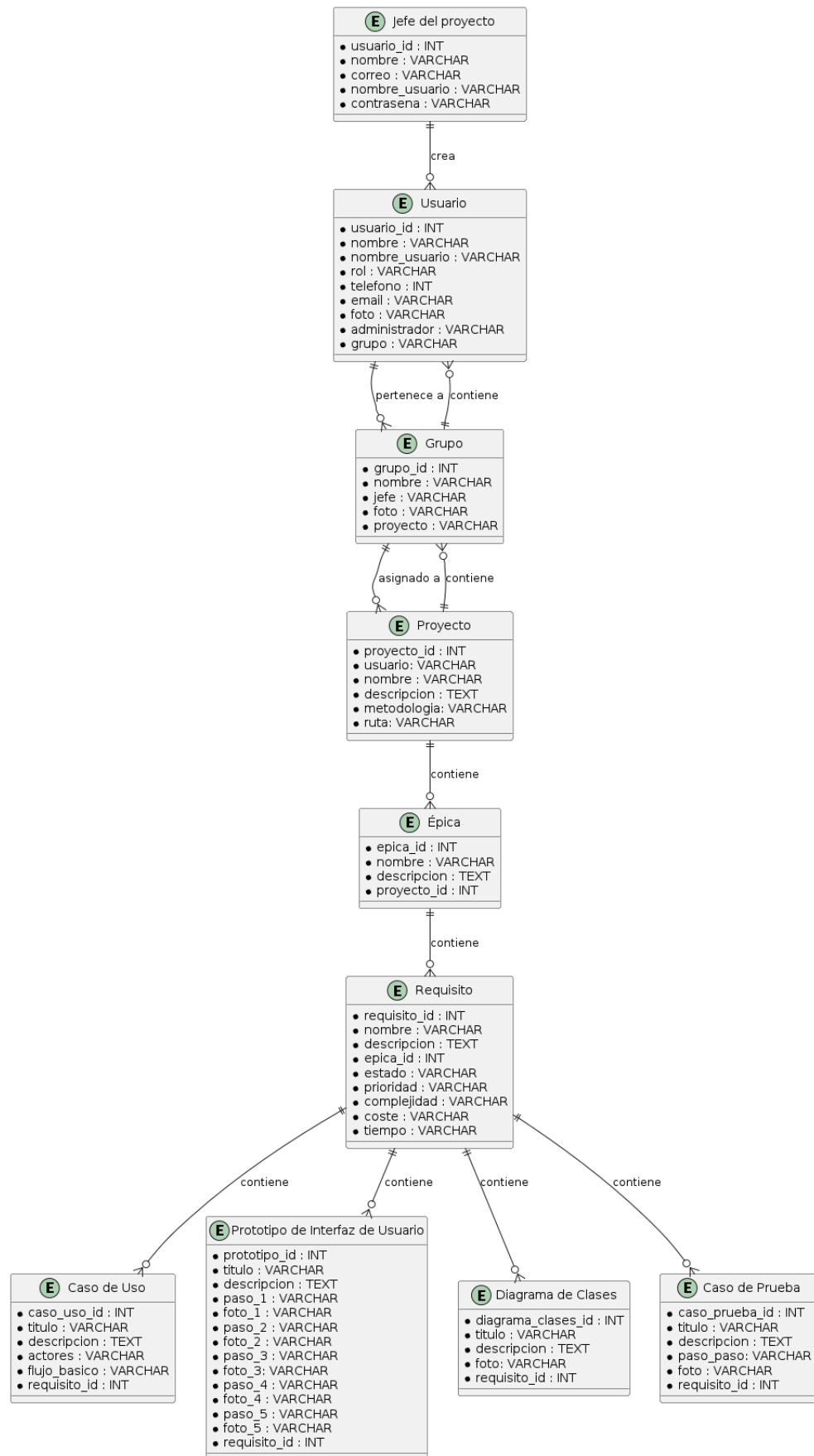


Ilustración 20: Diagrama entidad/relación

5. DISEÑO

En este capítulo estudiaremos el diseño en el que se ha basado la implementación, incluyendo su arquitectura lógica, diagrama de clases, base de datos e interfaz de usuario.

5.1. Arquitectura lógica

La arquitectura lógica de la aplicación está diseñada para organizar y gestionar los distintos componentes del sistema de manera clara y eficiente. A continuación, se detalla cómo se interrelacionan estos componentes:

- **Usuarios:** La aplicación distingue entre dos tipos de usuarios: administradores (jefes) y usuarios regulares. Los administradores tienen permisos ampliados, lo que les permite crear, modificar y eliminar otros usuarios, así como gestionar proyectos y grupos de trabajo. Los usuarios regulares tienen un conjunto de permisos más limitado, basado en las tareas asignadas dentro de sus grupos (roles).
- **Grupos:** Los usuarios pueden ser asignados a un grupo de trabajo. Estos grupos facilitan la organización del trabajo colaborativo dentro de los proyectos. Cada grupo tiene un nombre único y una lista de miembros.
- **Proyectos:** Los proyectos son la unidad principal de trabajo en la aplicación. Cada proyecto contiene un grupo de trabajo y está compuesto por varias épicas. Los proyectos tienen atributos como nombre, descripción y estado, y pueden ser creados, modificados o eliminados por los administradores.
- **Épicas:** Las épicas son grandes bloques de trabajo dentro de un proyecto. Cada épica está formada por un conjunto de requisitos que deben ser cumplidos para completar esa parte del proyecto. Las épicas facilitan la organización del trabajo en tareas manejables y enfocadas.
- **Requisitos:** Los requisitos son los componentes básicos del trabajo dentro de una épica. Cada requisito puede tener diferentes subcomponentes como casos de uso, prototipos de interfaz de usuario, diagramas de clases y casos de prueba. Estos subcomponentes ayudan a detallar y **clarificar la trazabilidad** para cumplir con cada requisito. Además, los requisitos tienen un estado que indica su progreso y cumplimiento.

5.2. Arquitectura física

Para asegurar el correcto funcionamiento de la arquitectura lógica descrita, es necesario contar con los siguientes elementos de hardware:

- **Servidor local:** Durante la fase de desarrollo y pruebas, se utiliza un servidor XAMPP. XAMPP es un paquete de software que incluye Apache como servidor web, MySQL como sistema de gestión de bases de datos y PHP como lenguaje de script del lado del servidor. Este entorno nos permite probar la aplicación localmente antes de su despliegue en producción.
- **Equipo de desarrollo:** Los desarrolladores necesitan un ordenador con suficiente capacidad para ejecutar Visual Studio Code y XAMPP simultáneamente. Un equipo con un procesador moderno, suficiente memoria RAM y almacenamiento es ideal para este propósito.
- **Servidor de producción:** Una vez completada la fase de desarrollo y pruebas, la aplicación se despliega en un servidor de producción. Este servidor debe soportar PHP y MySQL para asegurar la funcionalidad completa de la aplicación.

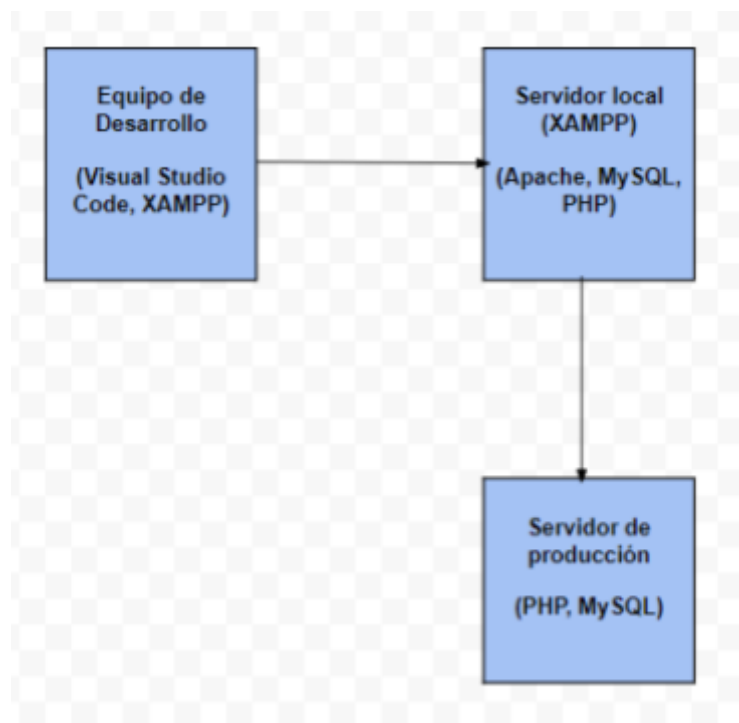
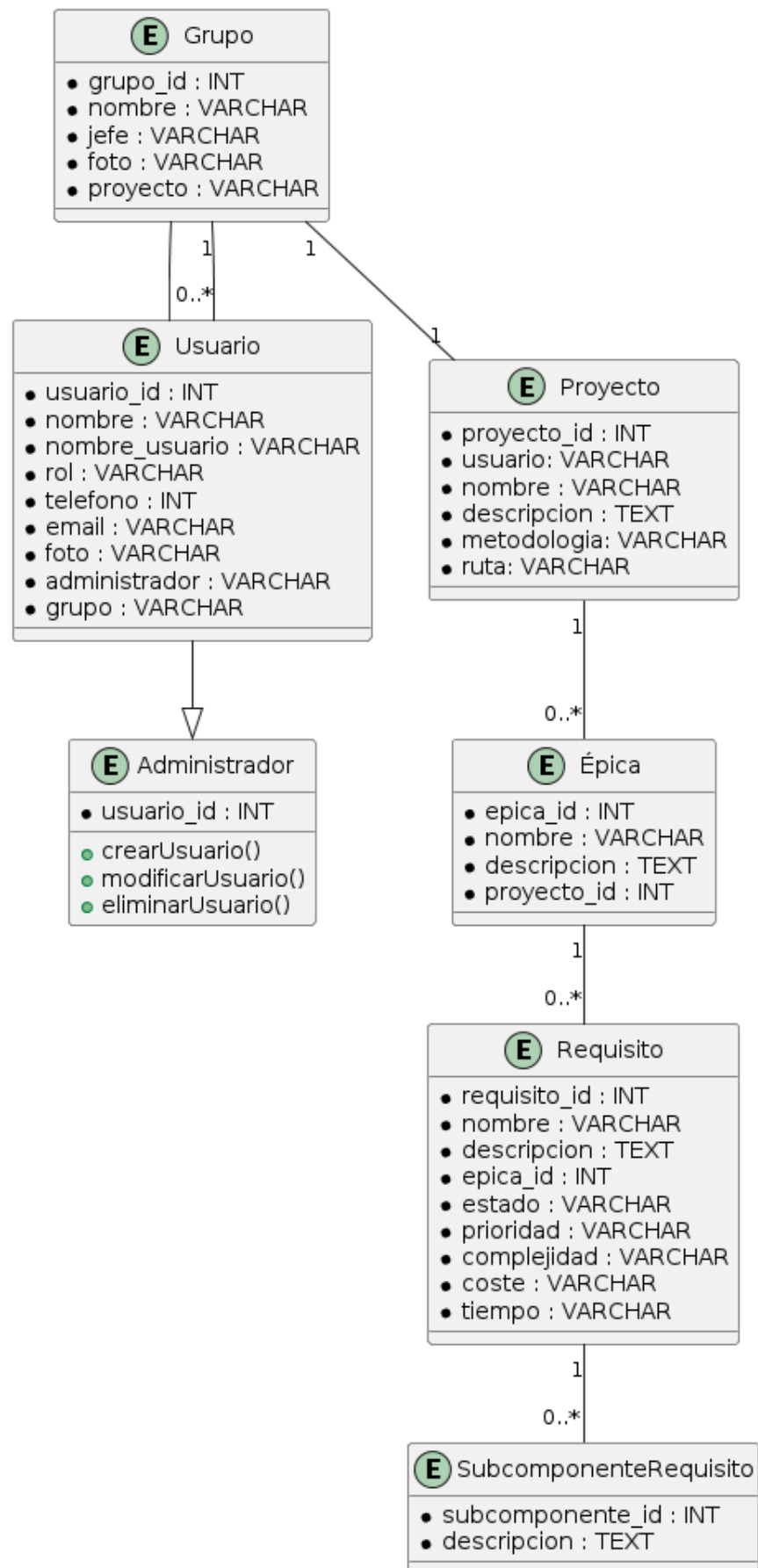


Ilustración 21: Arquitectura física

5.3. Diagrama de clases

El diagrama de clases proporciona una representación detallada de la arquitectura lógica de la aplicación a nivel de programación. A continuación se describen las principales clases y sus relaciones:

- **Administrador:** La clase Administrador hace referencia a el usuario registrado, el cual será el Jefe del Proyecto. Los métodos asociados a esta clase permiten la creación, modificación y eliminación de usuarios, así como la gestión de sus permisos y asignaciones a grupos.
- **Usuario:** La clase Usuario incluye atributos como nombre, correo electrónico, rol, grupo asociado etc... Esta clase tiene una asociación de uno a muchos con la clase Grupos.
- **Grupo:** La clase Grupo contiene atributos como nombre del grupo, jefe, proyecto asociado, etc... Esta clase tiene una asociación de uno a uno con la clase Proyecto.
- **Proyecto:** La clase Proyecto incluye atributos como nombre del proyecto, descripción. Los métodos permiten la gestión completa de los proyectos, incluyendo su creación, modificación y eliminación.
- **Épica:** La clase Épica contiene atributos como nombre, proyecto asociado. Los métodos permiten la gestión de las épicas dentro de un proyecto.
- **Requisito:** La clase Requisito incluye atributos como descripción, estado, épica asociada y una lista de subcomponentes (casos de uso, prototipos, etc.). Los métodos permiten la creación, modificación y eliminación de requisitos, así como la gestión de su estado y subcomponentes.



*Ilustración 22: Diagrama de clases***5.4. Diseño de la base de datos**

La base de datos es un componente crítico de la aplicación, ya que almacena toda la información necesaria para su funcionamiento. A continuación se describe su diseño:

- **Tablas principales:** La base de datos incluye tablas para Usuarios, Grupos, Proyectos, Épicas y Requisitos. Cada tabla está diseñada para almacenar la información relevante de manera eficiente.

Relaciones:

- Usuarios y Grupos: Un usuario pertenece a un único grupo, y cada grupo puede contener múltiples usuarios. Esta relación se gestiona mediante una tabla intermedia que enlaza usuarios y grupos.
- Grupos y Proyectos: Un grupo puede estar asignado a un único proyecto, y cada proyecto a un único grupo. Esta relación también se gestiona mediante una tabla intermedia.
- Proyectos y Épicas: Cada proyecto contiene múltiples épicas. La tabla de épicas incluye una referencia al proyecto al que pertenece.
- Épicas y Requisitos: Cada épica está formada por múltiples requisitos. La tabla de requisitos incluye una referencia a la épica a la que pertenece.

5.4.1. Diccionario de datos

En este apartado se explicarán los diferentes atributos de cada tabla de la base de datos, con sus diferentes restricciones, que se desarrollarán y utilizarán para la creación de la aplicación.

TABLA USUARIOS

Descripción		Esta tabla contiene la información de los usuarios registrados para ser administradores.			
Campo		Tipo	Tamaño	Clave	Restricciones
id		INT	11	PRIMARY	Not null Auto increment >0
nombre_completo		VARCHAR	50		Not null
usuario		VARCHAR	50	UNIQUE	Not null Unique

contraseña	VARCHAR	50		Not null
email	VARCHAR	50		Not null

TABLA USUARIOS_BD**Descripción**

Esta tabla contiene la información de los usuarios creados por el administrador.

Campo	Tipo	Tamaño	Clave	Restricciones
usuario_id	INT	11	PRIMARY	Not null Auto increment >0
nombre	VARCHAR	50		Not null
usuario	VARCHAR	50	UNIQUE	Not null Unique
rol	VARCHAR	50		Not null
email	VARCHAR	50		Not null
telefono	INT	50		Not null
ruta	VARCHAR	120		Not null
administrador	VARCHAR	50		Not null
grupo	VARCHAR	50	UNIQUE	Not null Unique

TABLA GRUPOS_BD**Descripción**

Esta tabla contiene la información de los grupos de trabajo creados por el administrador.

Campo	Tipo	Tamaño	Clave	Restricciones
id	INT	11	PRIMARY	Not null Auto increment >0
nombre	VARCHAR	50		Not null
jefe	VARCHAR	50	UNIQUE	Not null Unique
ruta	VARCHAR	200		Not null
grupo	VARCHAR	50	UNIQUE	Not null Unique

TABLA PROYECTOS_BD**Descripción**

Esta tabla contiene la información de los proyectos creados por el administrador.

Campo

	Tipo	Tamaño	Clave	Restricciones
id	INT	11	PRIMARY	Not null Auto increment >0
nombre	VARCHAR	50		Not null
usuario	VARCHAR	50	UNIQUE	Not null Unique
descripcion	VARCHAR	50		Not null
metodologia	VARCHAR	50		Not null
ruta	VARCHAR	200		Not null

TABLA ÉPICAS_BD**Descripción**

Esta tabla contiene la información de las épicas creadas por el administrador.

Campo

	Tipo	Tamaño	Clave	Restricciones
usuario_id	INT	11	PRIMARY	Not null Auto increment >0
nombre	VARCHAR	50	UNIQUE	Not null Unique
descripcion	VARCHAR	500		Not null
proyecto_id	INT	11	UNIQUE	Not null Unique

TABLA REQUISITOS_BD**Descripción**

Esta tabla contiene la información de los requisitos creados por el administrador.

Campo

	Tipo	Tamaño	Clave	Restricciones
id	INT	11	PRIMARY	Not null Auto increment >0

nombre	VARCHAR	50		Not null
descripcion	VARCHAR	500	UNIQUE	Not null Unique
proyecto_id	INT	11	UNIQUE	Not null Unique
epica	INT	11	UNIQUE	Not null Unique
estado	VARCHAR	50		Not null
prioridad	VARCHAR	120		Not null
complejidad	VARCHAR	50		Not null
coste	INT	11		Not null
tiempo	INT	11		Not null

TABLA CASODEUSO_BD

Descripción	Esta tabla contiene la información de los casos de uso creados por el administrador para un requisito.			
Campo	Tipo	Tamaño	Clave	Restricciones
id	INT	11	PRIMARY	Not null Auto increment >0
título	VARCHAR	50	UNIQUE	Not null Unique
descripcion	VARCHAR	500		Not null
actores	VARCHAR	500		Not null
flujo_basico	VARCHAR	1000		Not null
requisito_id	VARCHAR	50	UNIQUE	Not null Unique

TABLA _PROTOTIPO_BD**Descripción**

Esta tabla contiene la información de los prototipos de interfaz de usuario creados por el administrador para un requisito.

Campo

	Tipo	Tamaño	Clave	Restricciones
id	INT	11	PRIMARY	Not null Auto increment >0
título	VARCHAR	50	UNIQUE	Not null Unique
descripcion	VARCHAR	500		Not null
paso_1	VARCHAR	50		Not null
foto1	VARCHAR	50		Not null
paso_2	VARCHAR	50		Not null
foto2	VARCHAR	50		Not null
paso_3	VARCHAR	50		Not null
foto3	VARCHAR	50		Not null
paso_4	VARCHAR	50		Not null
foto4	VARCHAR	50		Not null
paso_5	VARCHAR	50		Not null
foto5	VARCHAR	50		Not null
requisito_id	VARCHAR	50	UNIQUE	Not null Unique

TABLA PASO_PASO_BD**Descripción**

Esta tabla contiene la información de los casos de prueba creados por el administrador para un requisito.

Campo

	Tipo	Tamaño	Clave	Restricciones
--	-------------	---------------	--------------	----------------------

id	INT	11	PRIMARY	Not null Auto increment >0
título	VARCHAR	50	UNIQUE	Not null Unique
descripcion	VARCHAR	500		Not null
paso_paso	VARCHAR	1000		Not null
foto	VARCHAR	500		Not null
requisito_id	VARCHAR	50	UNIQUE	Not null Unique

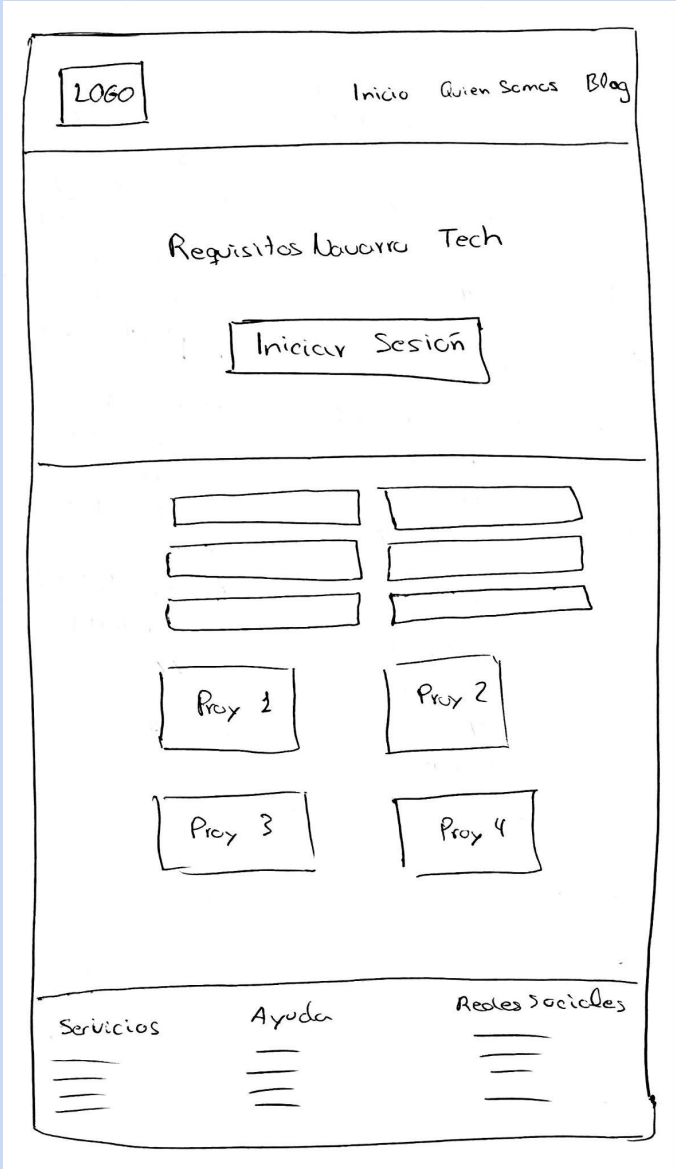
TABLE DIAGRAMA_BD

Descripción	Esta tabla contiene la información de los diagramas de clases creados por el administrador para un requisito.			
Campo	Tipo	Tamaño	Clave	Restricciones
id	INT	11	PRIMARY	Not null Auto increment >0
título	VARCHAR	50	UNIQUE	Not null Unique
descripcion	VARCHAR	500		Not null
foto	VARCHAR	500		Not null
requisito_id	VARCHAR	50	UNIQUE	Not null Unique

5.5. Diseño de la interfaz de usuario

El diseño de la interfaz de usuario (UI) se ha centrado en proporcionar una experiencia intuitiva y eficiente para los usuarios. A continuación se describen los principales aspectos del diseño de la UI elegido en forma de mockups de pantallas web para las diversas páginas que componen las vistas de la aplicación:

Mockup interfaz página inicial

Descripción	En esta vista aparecerán los datos e información de la aplicación, con un apartado central para ir a la pantalla de registro e inicio de sesión.
Activación	Esta pantalla aparecerá siempre que inicies la aplicación.
Boceto	

Eventos

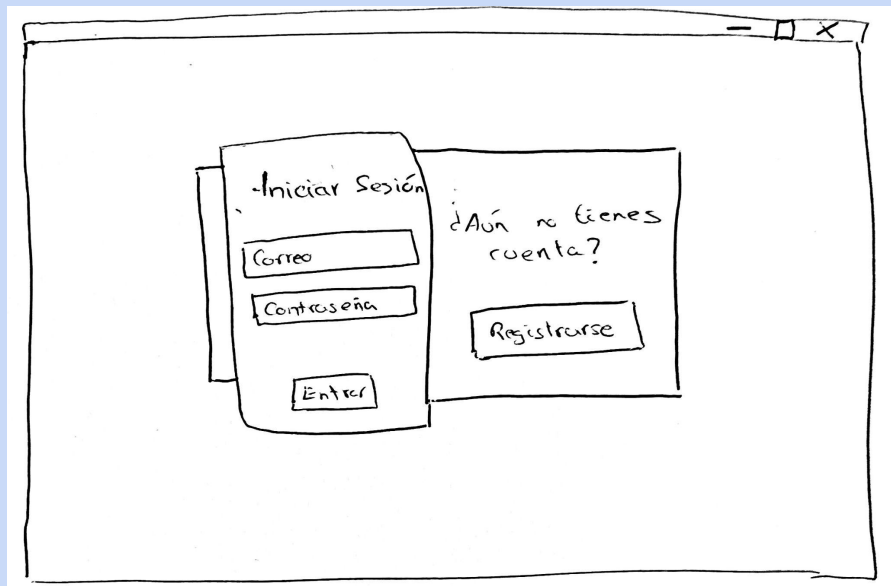
- Enlaces del “**header**” : enlaces como inicio, quiénes somos, blog o contacto, mostrarán información relacionada con la aplicación.
- Enlace “**Iniciar Sesión**” : nos llevará a la pantalla de registro e inicio de sesión.
- Cuadros con utilidades de la aplicación y antiguos trabajos realizados
- Footer con información, servicios o redes sociales.

Requisitos que satisface**Mockup interfaz página de registro****Descripción**

En esta vista aparecerá la pantalla de registro e inicio de sesión para entrar a la aplicación.

Activación

Esta pantalla aparecerá pulsando el botón de inicio de sesión de la pantalla anterior o pulsando el botón de cerrar sesión que aparecerá más adelante.

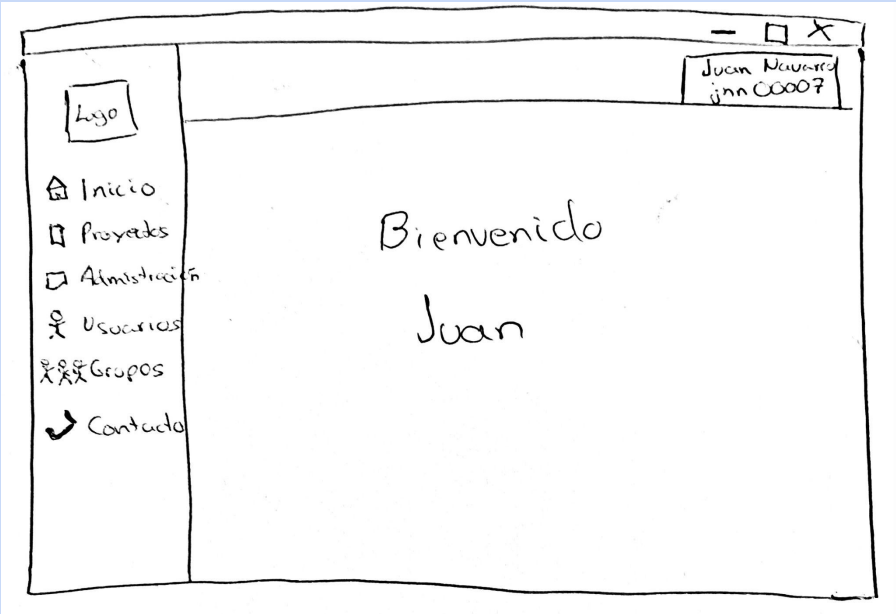
Boceto**Eventos**

- Botón de “**Registrarse**” : botón que nos finaliza el registro de un nuevo administrador y nos pasará al apartado de inicio de sesión si todos los datos son correctos.
- Botón de “**Iniciar Sesión**” : botón que nos llevará a la página principal de la aplicación como administrador registrados si todos los datos son correctos.

Requisitos que satisface

EPIC-01 y EPIC-02

Mockup interfaz página inicial

Descripción	En esta vista aparecerán los datos e información de la aplicación, con un apartado central para ir a la pantalla de registro e inicio de sesión.
Activación	Esta pantalla aparecerá siempre que inicies la aplicación.
Boceto	
Eventos	● Menú lateral : al pulsarlo aparecerán las distintas acciones que se pueden realizar en la aplicación. ○ Botón “Inicio” : nos volverá a la pantalla donde estamos. ○ Botón “Proyectos” : nos llevará a la pantalla donde se visualizarán los proyectos creados. ○ Botón “Administración” : Al pulsarlo aparecen 3 nuevos botones para: ■ “Crear proyecto”: La creación de un nuevo proyecto. ■ “Modificar proyecto”: La modificación de un proyecto. ■ “Eliminar proyecto”: La eliminación de un proyecto. ○ Botón “Usuarios” : Al pulsarlo aparecen 2 nuevos botones para: ■ “Crear usuario”: La pantalla para la creación de un nuevo usuario. ■ “Ver usuarios”: Ir a la pantalla donde se visualizarán los usuarios creados. ○ Botón “Grupos” : Al pulsarlo aparecen 2 nuevos botones para: ■ “Crear grupo”: La pantalla para la creación de un nuevo grupo de trabajo. ■ “Ver grupos”: Ir a la pantalla donde se visualizarán los grupos de trabajo creados. ○ Botón “Contacto” : nos llevará a la pantalla donde hay

Requisitos
que
satisface

información de contacto por si hay algún fallo.

- Botón **Usuario Registrado**: aparece el nombre completo y el nombre de usuario del administrador registrado que está utilizando la aplicación.
 - Al pulsar sobre él nos aparece un nuevo boton "**Cerrar sesión**" que nos cierra la sesión de ese administrador y nos lleva a la pantalla de registro.

Mockup interfaz página Crear Proyecto

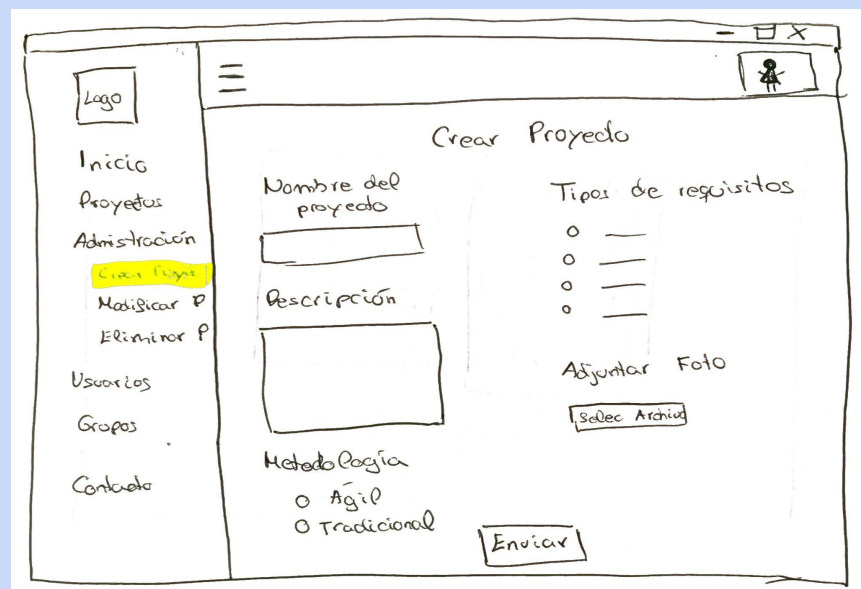
Descripción

En esta vista aparecerá el formulario para la creación de un nuevo proyecto

Activación

Esta pantalla aparecerá siempre que pulses el botón "Crear proyecto".

Boceto



Eventos

- El **menú lateral** será el mismo, explicado anteriormente.
- El **recuadro de usuario registrado** será el mismo, explicado anteriormente.
- Formulario para la creación de un nuevo proyecto:
 - Cuadro de texto para "Nombre del proyecto".
 - Cuadro de texto para "Descripción".
 - Tipo Radio para elegir "Metodología".
 - Tipo Checkbox para elegir "Tipos de requisitos".
 - Tipo File para adjuntar "Imagen".
- Botón "**Enviar**" para confirmar la creación de un nuevo proyecto

Requisitos
que
satisface

FRQ-12

Mockup interfaz página Modifica Proyecto

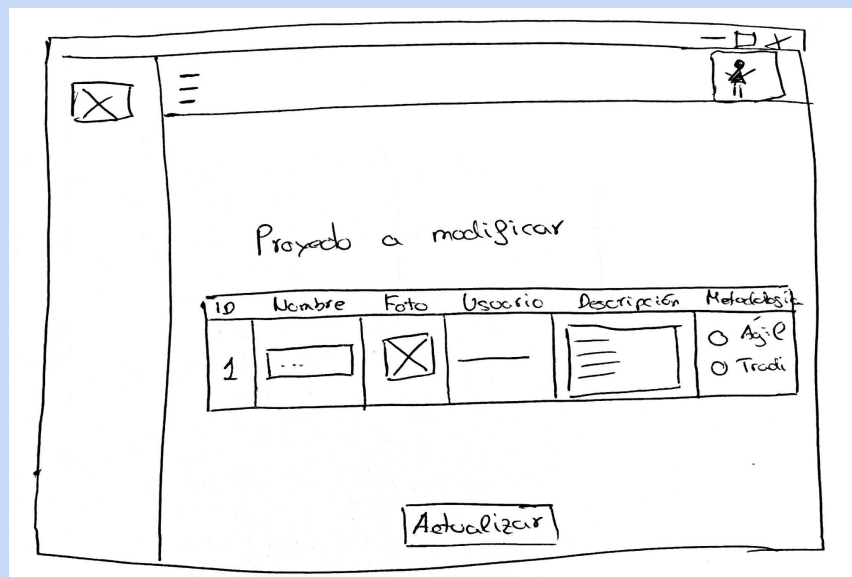
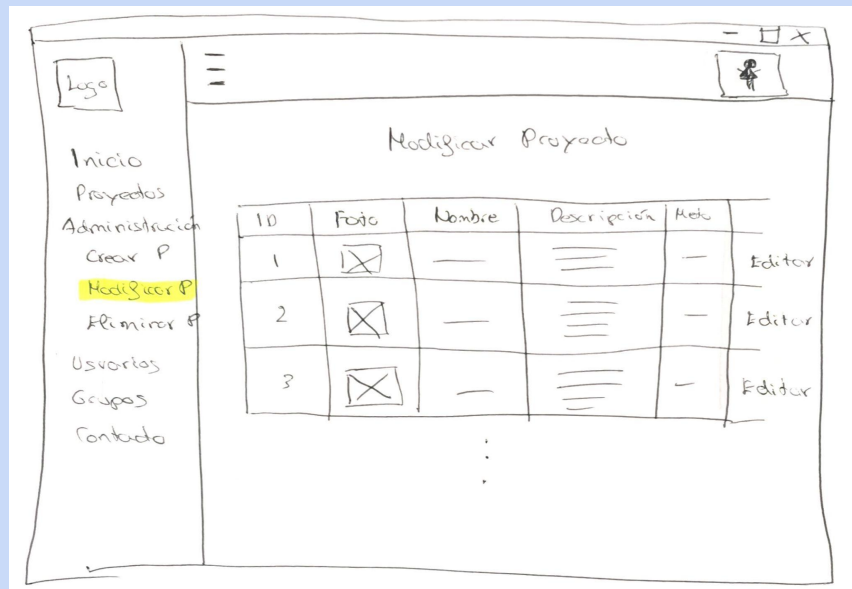
Descripción

En esta vista aparecerá el formulario para la modificación de un proyecto

Activación

Esta pantalla aparecerá siempre que pulses el botón “Modificar proyecto”.

Boceto



Eventos

- El **menú lateral** será el mismo, explicado anteriormente.
- El **recuadro de usuario registrado** será el mismo, explicado anteriormente.
- Aparecerán todos los proyecto creados con un botón a la derecha “**Editar**” que al pulsarlo nos llevará al siguiente formulario:
 - Formulario para la modificación de un proyecto:
 - Cuadro de texto para modificar “Nombre del proyecto”.
 - Cuadro de texto para modificar “Descripción”.

Requisitos
que
satisface

FRQ-13

- Tipo Radio para modificar “Metodología”.
- Botón “**Actualizar**” para confirmar la modificación del proyecto

Mockup interfaz página Elimina Proyecto

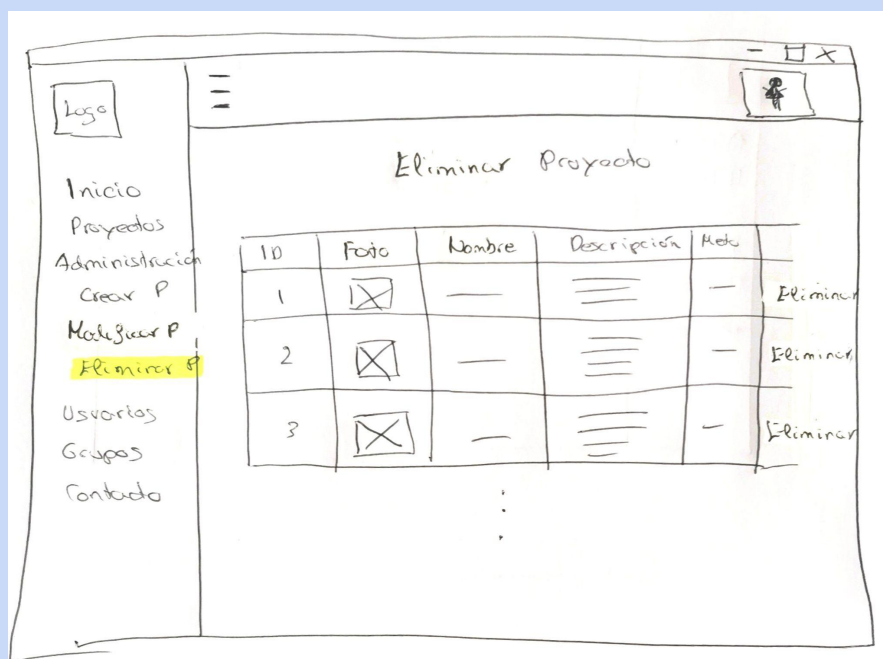
Descripción

En esta vista aparecerá el formulario para la eliminación de un proyecto

Activación

Esta pantalla aparecerá siempre que pulses el botón “Eliminar proyecto”.

Boceto



Eventos

- El **menú lateral** será el mismo, explicado anteriormente.
- El **recuadro de usuario registrado** será el mismo, explicado anteriormente.
- Aparecerán todos los proyecto creados con un botón a la derecha “**Eliminar**” que al pulsarlo nos saltará un mensaje de confirmación para la eliminación del proyecto.

Requisitos
que
satisface

FRQ-14

Mockup interfaz página Visualizar Proyectos

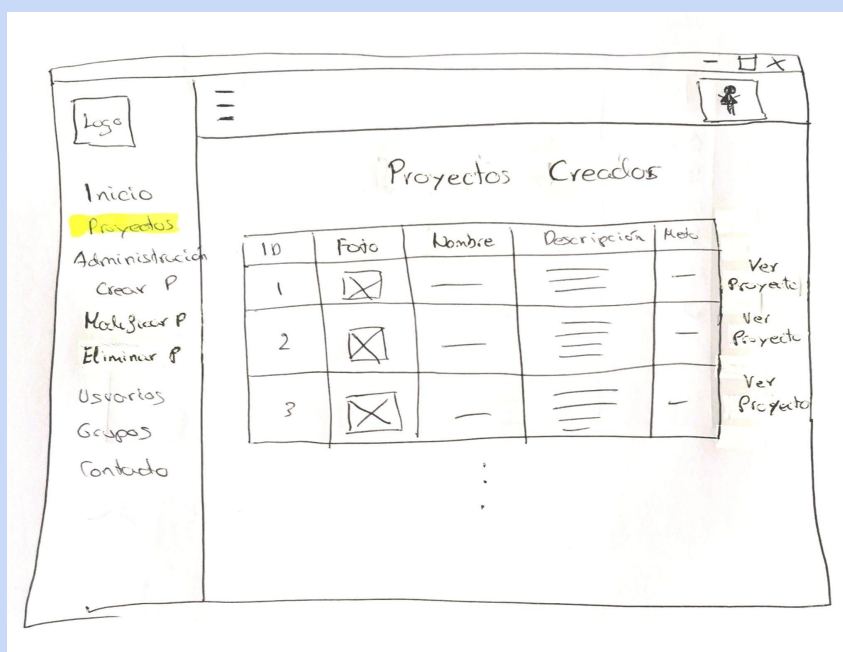
Descripción

En esta vista aparecerá la visualización de los proyectos creados.

Activación

Esta pantalla aparecerá siempre que pulses el botón “Proyectos”.

Boceto



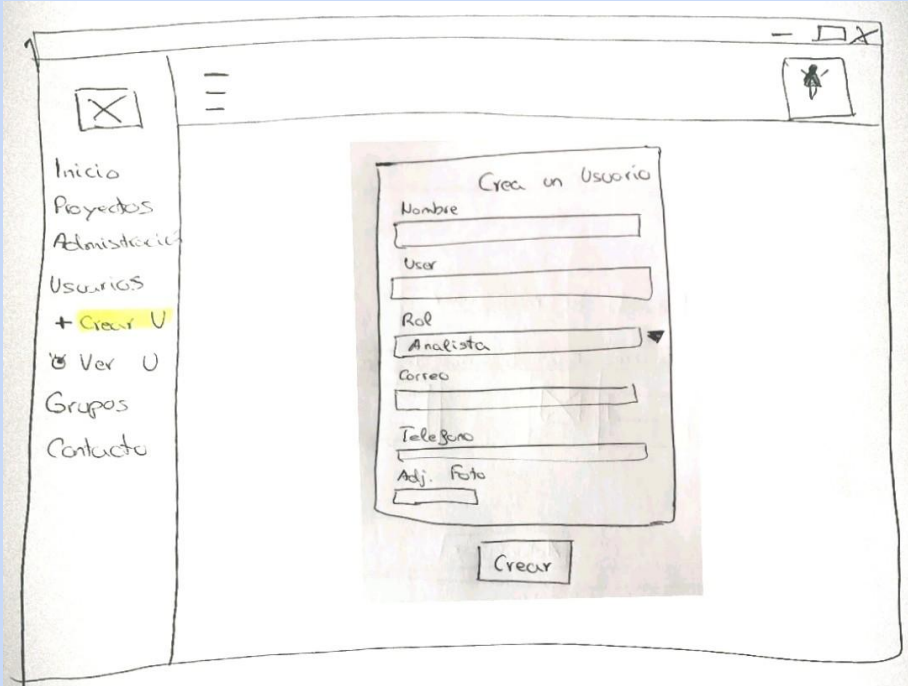
Eventos

- El **menú lateral** será el mismo, explicado anteriormente.
- El **recuadro de usuario registrado** será el mismo, explicado anteriormente.
- Aparecerán todos los proyecto creados con un botón a la derecha “**Ver Proyecto**” que al pulsarlo nos llevará a la pantalla de información del proyecto

Requisitos que satisface

FRQ-15

Mockup interfaz página Crear Usuario

Descripción	En esta vista aparecerá el formulario para la creación de un nuevo usuario.
Activación	Esta pantalla aparecerá siempre que pulses el botón “Crear usuario”.
Boceto	
Eventos	• El menú lateral será el mismo, explicado anteriormente. • El recuadro de usuario registrado será el mismo, explicado anteriormente. • Formulario para la creación de un nuevo usuario: ○ Cuadro de texto para “Nombre completo”. ○ Cuadro de texto para “User”. ○ Tipo Radio para elegir “Rol del usuario”. ○ Cuadro de texto para “Correo electrónico” ○ Cuadro de texto para “Teléfono” ○ Tipo File para adjuntar “Imagen” • Botón “Crear” para confirmar la creación de un nuevo proyecto
Requisitos que satisface	FRQ-01 y FRQ-04

Mockup interfaz página Visualizar Usuarios

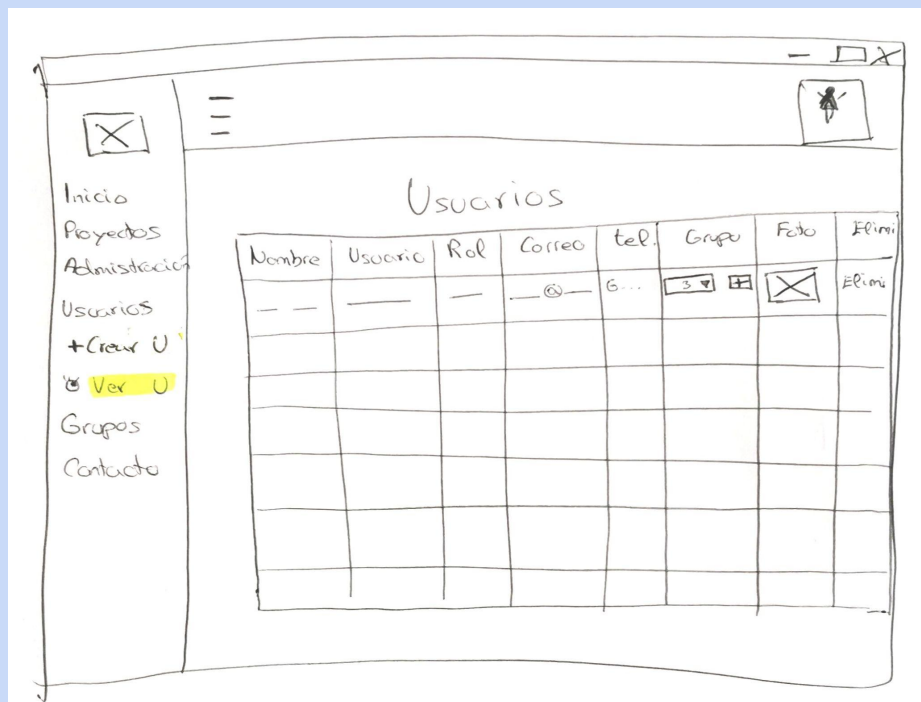
Descripción

En esta vista aparecerá la visualización de los usuarios creados.

Activación

Esta pantalla aparecerá siempre que pulses el botón “Ver usuarios”.

Boceto



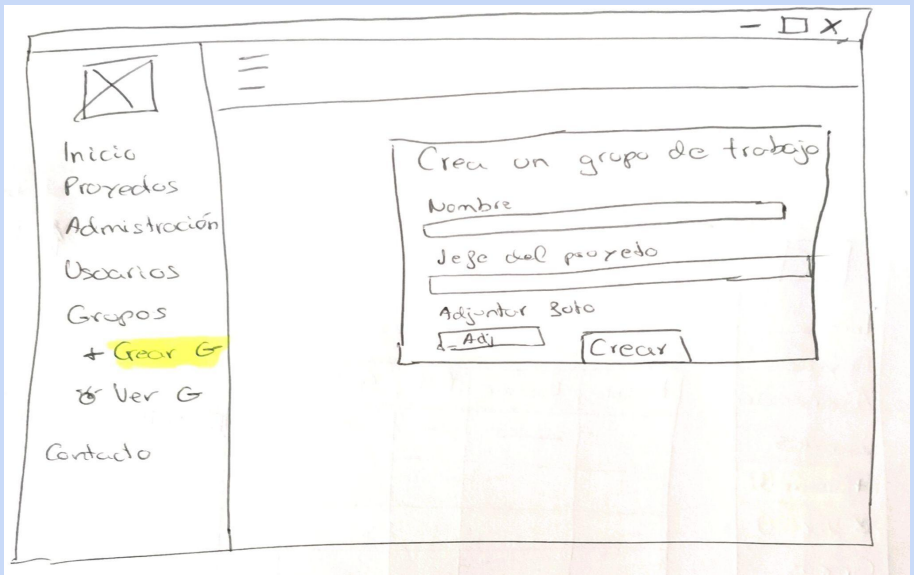
Eventos

- El **menú lateral** será el mismo, explicado anteriormente.
- El **recuadro de usuario registrado** será el mismo, explicado anteriormente.
- Aparecerán todos los usuarios creados con un botón a la derecha “**Eliminar**” que al pulsarlo nos saltará un mensaje de confirmación para la eliminación del usuario.
- Botón “**Añadir**” : un desplegable con los grupos de trabajo creados, el cual podemos elegir el grupo de trabajo al que queremos añadir al usuario.

Requisitos que satisface

FRQ-03, FRQ-05 y FRQ-06

Mockup interfaz página Crear Grupo de trabajo

Descripción	En esta vista aparecerá el formulario para la creación de un nuevo grupo de trabajo.
Activación	Esta pantalla aparecerá siempre que pulses el botón “Crear grupo”.
Boceto	
Eventos	• El menú lateral será el mismo, explicado anteriormente. • El recuadro de usuario registrado será el mismo, explicado anteriormente. • Formulario para la creación de un nuevo grupo de trabajo: ○ Cuadro de texto para “Nombre del grupo de trabajo”. ○ Cuadro de texto para “Jefe del proyecto”. ○ Tipo File para adjuntar “Imagen” • Botón “Crear” para confirmar la creación de un nuevo proyecto
Requisitos que satisface	FRQ-07

Mockup interfaz página Visualizar Grupos de trabajo

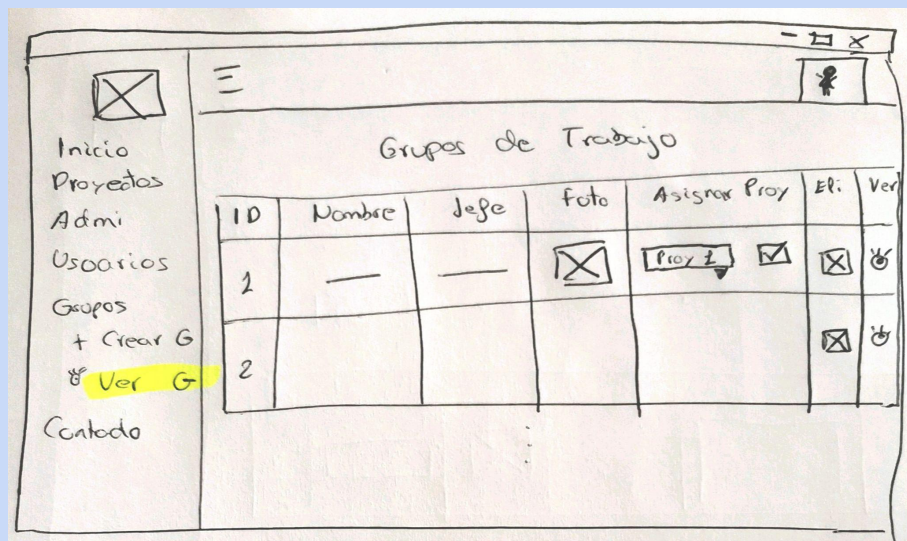
Descripción

En esta vista aparecerá la visualización de los grupos de trabajo creados.

Activación

Esta pantalla aparecerá siempre que pulses el botón “Ver grupos”.

Boceto



Eventos

- El **menú lateral** será el mismo, explicado anteriormente.
- El **recuadro de usuario registrado** será el mismo, explicado anteriormente.
- Aparecerán todos los grupos de trabajo creados con un botón a la derecha “**Eliminar**” que al pulsarlo nos saltará un mensaje de confirmación para la eliminación del grupo de trabajo.
- Botón “**Asignar**” : un desplegable con los proyectos creados, el cual podemos elegir el proyecto al que queremos añadir al grupo de trabajo.
- Botón “**Ver Grupo**” : un desplegable con los proyectos creados, el cual podemos elegir el proyecto al que queremos añadir al grupo de trabajo.

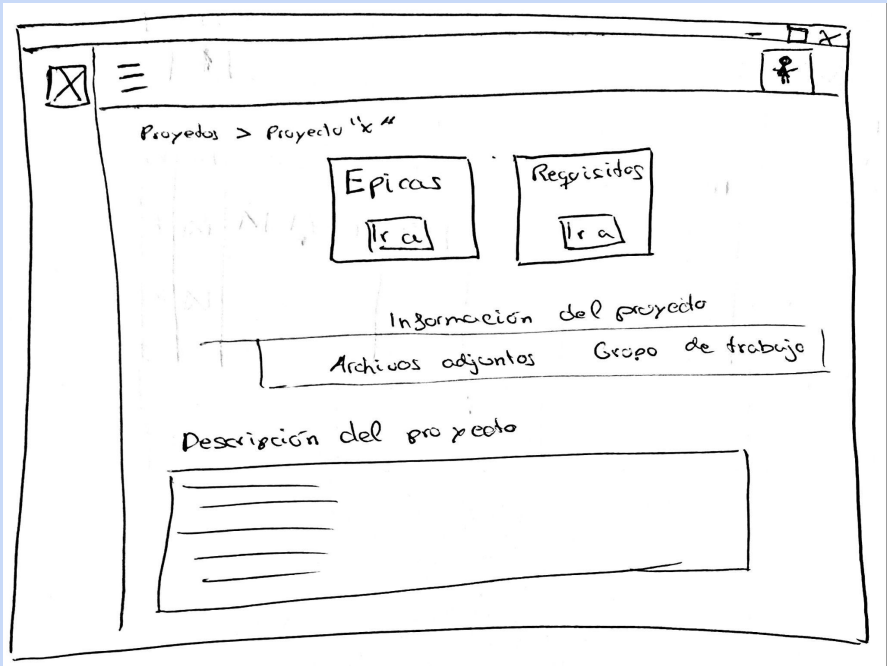
Requisitos que satisface

FRQ-08, FRQ-09 y FRQ-11

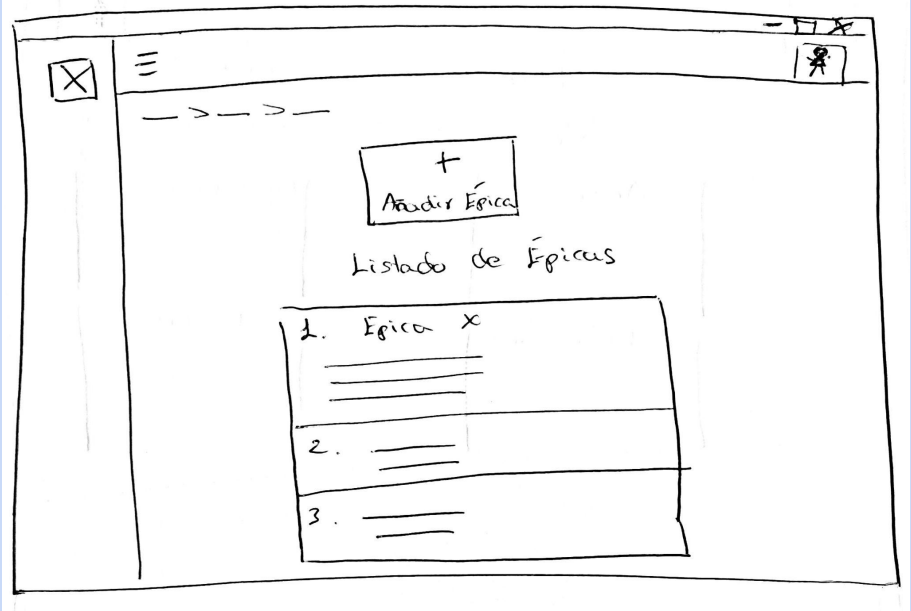
Mockup interfaz página Visualizar Grupo de trabajo

Descripción	En esta vista aparecerá la visualización del grupo de trabajo seleccionado.
Activación	Esta pantalla aparecerá siempre que pulses el botón “Ver grupos”, dentro de la pantalla visualización de grupos.
Boceto	
Eventos	• El menú lateral será el mismo, explicado anteriormente. • El recuadro de usuario registrado será el mismo, explicado anteriormente. • Aparecerá el jefe del proyecto asignado a ese grupo. • Aparecerán todos los usuarios/trabajadores asignados a ese grupo, separados por roles y con la información de cada uno
Requisitos que satisface	FRQ-10

Mockup interfaz página Información proyecto

Descripción	En esta vista aparecerá la información del proyecto seleccionado
Activación	Esta pantalla aparecerá siempre que pulses el botón “Ver proyecto”, de la pantalla de visualización de proyectos.
Boceto	
Eventos	• El menú lateral será el mismo, explicado anteriormente. • El recuadro de usuario registrado será el mismo, explicado anteriormente. • Botón “Ir a”: para la creación de una nueva épica para el proyecto: • Botón “Ir a”: para la creación de un nuevo requisito para el proyecto: • Enlace “Archivos adjuntos” para visualizar los archivos adjuntos del proyecto. • Enlace “Grupo de trabajo” para visualizar el grupo de trabajo asignado al proyecto. • Cuadro de texto para ver “Descripción del proyecto”.
Requisitos que satisface	FRQ-16

Mockup interfaz página Crear Épica

Descripción	En esta vista aparecerá el formulario para la creación de una nueva épica y para la visualización de las épicas creadas.
Activación	Esta pantalla aparecerá siempre que pulses el botón “Ir a”, del apartado de épicas o al pulsar el atajo de enlaces.
Boceto	
Eventos	• El menú lateral será el mismo, explicado anteriormente. • El recuadro de usuario registrado será el mismo, explicado anteriormente. • Atajo de enlaces donde aparecen todo el camino seguido para llegar allí “Proyectos > Proyecto “EverSync” > Épicas” • Botón “Añadir épica”: Formulario para la creación de una nueva épica: ○ Cuadro de texto para “Título de la épica”. ○ Cuadro de texto para “Descripción”. ○ Botón “Crear” para confirmar la creación de la épica. • Listado de todas las épicas de ese proyecto con su descripción asociada.
Requisitos que satisface	FRQ-17 y FRQ-18

Mockup interfaz página Crear Requisito

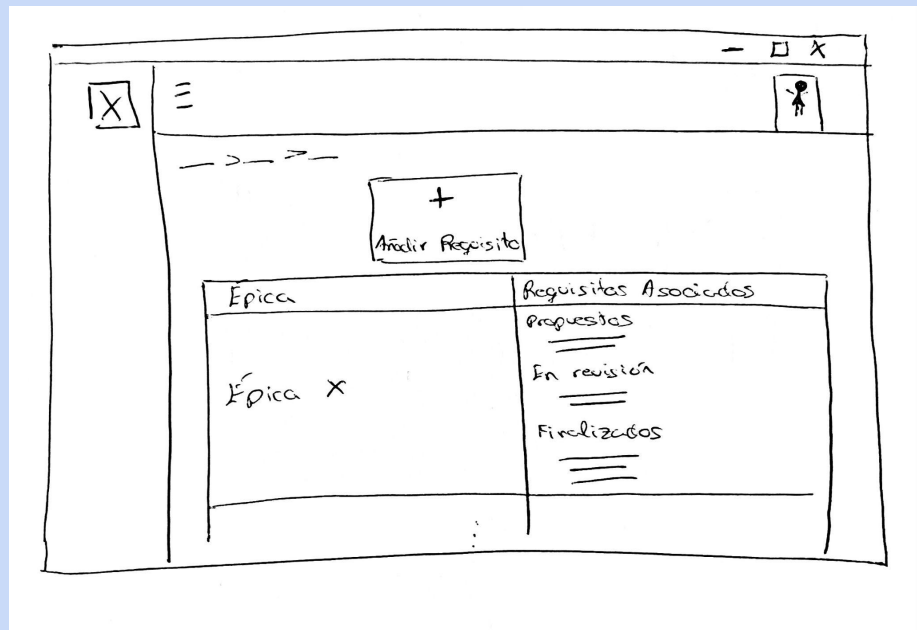
Descripción

En esta vista aparecerá el formulario para la creación de un nuevo requisito y para la visualización de los requisitos ya creados.

Activación

Esta pantalla aparecerá siempre que pulses el botón “Ir a”, del apartado de requisitos o al pulsar el atajo de enlaces.

Boceto



Eventos

- El **menú lateral** será el mismo, explicado anteriormente.
- El **recuadro de usuario registrado** será el mismo, explicado anteriormente.
- Atajo de enlaces donde aparecen todo el camino seguido para llegar allí “Proyectos > Proyecto “EverSync” > Épicas > Requisitos”
- Botón “Añadir requisito”: Formulario para la creación de un nuevo requisito:
 - Cuadro de texto para “Título del requisito”.
 - Cuadro de texto para “Descripción”.
 - Tipo Radio para elegir “Estado”.
 - Tipo Radio para elegir “Prioridad”.
 - Tipo Radio para elegir “Complejidad”.
 - Tipo Radio para elegir “Selección épica: seleccionar la épica a la que pertenecerá el requisito”.
 - Cuadro de texto para “Coste”.
 - Cuadro de texto para “Tiempo”.
 - Botón “**Crear**” para confirmar la creación del requisito.
- Listado de todos los requisitos asignados de ese proyecto con su descripción asociada, diferenciados por el estado en el que se encuentra el requisito y por la épica a la que pertenece.
- Cada requisito será un enlace que nos llevará a la trazabilidad del requisito en cuestión

Requisitos
que
satisface

FRQ-19, FRQ-20 y FRQ-21

Mockup interfaz página Visualizar Requisito

Descripción

En esta vista aparecerá el formulario para la creación de nuevos artefactos para el requisito, la visualización de la información del requisito y, lo más importante, la trazabilidad de este.

Activación

Esta pantalla aparecerá siempre que pulses el requisito al que quieras ir del apartado de requisitos o al pulsar el atajo de enlaces.

Boceto



Eventos

- El **menú lateral** será el mismo, explicado anteriormente.
- El **recuadro de usuario registrado** será el mismo, explicado anteriormente.
- Atajo de enlaces donde aparecen todo el camino seguido para llegar allí "Proyectos > Proyecto "EverSync" > Épicas > Requisitos > Requisito X"
- Cuadro con la información del requisito.
 - Formulario para el cambio de estado del requisito:
 - Tipo Radio para elegir "nuevo estado".
 - Botón "**Cambiar**" para confirmar el cambio de estado.
- Cuadro con toda la **trazabilidad** del requisito, la cual se irá actualizando en función de los artefactos que vayamos añadiendo..
- Formulario para la creación de los 4 tipos de artefactos disponibles: caso de uso, prototipo de interfaz de usuario, diagrama de clases y caso de prueba. Cada uno de ellos cuenta con un formulario con:
 - Cuadro de texto para "Título".

**Requisitos
que
satisface**

- Cuadro de texto para “Descripción”.
- Cuadro de texto para “Actores”.
- Cuadro de texto para “Flujo básico”.
- Al pulsar el botón “Crear”, se añadirá el artefacto creado y se actualizará la trazabilidad del requisito.

FRQ-22, FRQ-23, FRQ-24, FRQ-25, FRQ-26, FRQ-27 y FRQ-28

6. IMPLEMENTACIÓN

Tras la información obtenida en las fases de análisis y diseño, en esta sección, se detallará el proceso de implementación del proyecto, abordando las diversas etapas y técnicas empleadas para convertir los requisitos definidos en un producto de software funcional. En esta sección se describen las herramientas utilizadas, la estructura del proyecto y los detalles de implementación.

6.1. Herramientas utilizadas

En este apartado se describen las diferentes herramientas que se han utilizado para el desarrollo de la aplicación. Las tecnologías utilizadas para el proyecto se han explicado anteriormente en el apartado 3.4, por eso, en este apartado las mencionaremos y veremos un resumen de ellas:

- **Visual Studio Code:** Entorno de desarrollo profesional que nos ofrece una amplia gama de características y extensiones que facilitan la escritura de código en múltiples lenguajes.
- **MySQL:** Servidor de bases de datos relacionales desarrollado por Oracle.
- Lenguajes de programación:
 - **PHP** para la lógica del lado del servidor.
 - **CSS** para el diseño y la presentación de la interfaz de usuario.
 - **JavaScript** para la interactividad y la mejora de la experiencia del usuario.
- **XAMPP:** como servidor para la implementación y pruebas locales de la aplicación.

6.2. Arquitectura de componentes software

En esta sección, se presenta la arquitectura de la aplicación, detallando los componentes software que se han implementado.

- **Frontend:**
 - **Interfaz de Usuario (UI):** Desarrollada con HTML, CSS y JavaScript para garantizar una experiencia de usuario intuitiva y agradable.
 - **Componentes Visuales:** Elementos reutilizables como formularios, botones y menús, diseñados para mantener la consistencia visual y funcional en toda la aplicación.

- **Backend:**

- **Lógica del Servidor:** Implementada en PHP, encargada de manejar las solicitudes de los usuarios, procesar datos y comunicarse con la base de datos.
- **Controladores:** Gestionan la lógica de negocio y actúan como intermediarios entre la interfaz de usuario y la base de datos.

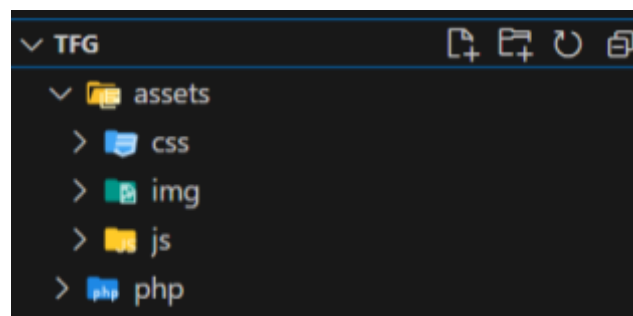


Ilustración 23:

A continuación, se procederá con una explicación de cada carpeta y de cada archivo que contiene:

- Carpeta “**assets**”: carpeta que contiene la parte de Frontend de la aplicación.
 - Carpeta “**css**”: carpeta que contiene las diferentes clases .css con todos los estilos de la aplicación, son las siguientes:

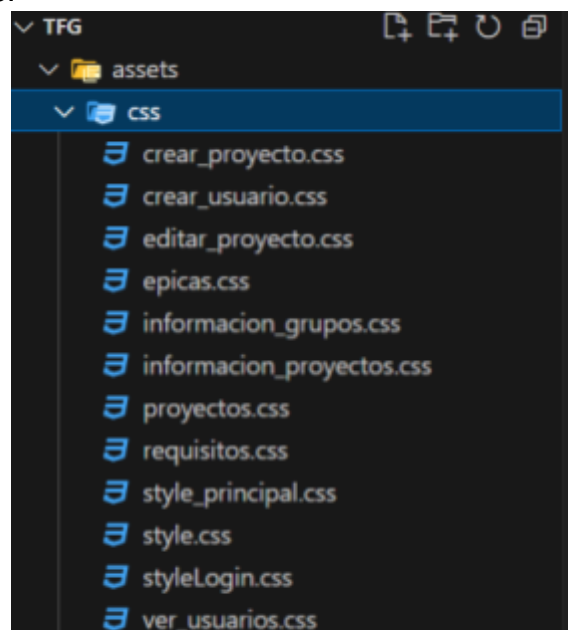


Ilustración 24: Contenido carpeta TFG

Clase	Descripción
crear_proyecto.css	Clase que configura los estilos de la pantalla de creación de un proyecto.
crear_usuario.css	Clase que configura los estilos de la pantalla de creación de un usuario.
editar_proyecto.css	Clase que configura los estilos de la pantalla de edición de un proyecto.
epicas.css	Clase que configura los estilos de la pantalla de creación y visualización de épicas.
informacion_proyectos.css	Clase que configura los estilos de la pantalla de información del proyecto seleccionado.
proyectos.css	Clase que configura los estilos de la pantalla de listado de todos los proyectos creados.
requisitos.css	Clase que configura los estilos de la pantalla de creación y visualización del listado de todos los requisitos creados.
style_principal.css	Clase que configura los estilos de la pantalla del menú lateral de la aplicación y del recuadro de usuario logueado.
style.css	Clase que configura los estilos de la pantalla de inicio de la aplicación.
styleLogin.css	Clase que configura los estilos de la pantalla de inicio de sesión y registro de la aplicación.
ver_usuarios.css	Clase que configura los estilos de la pantalla de listado de todos los usuarios creados.
informacion_grupos.css	Clase que configura los estilos de la pantalla de información del grupo seleccionado.

- Carpeta “**img**”: carpeta que contiene todas las imágenes que tiene la aplicación, además de las que se van añadiendo cuando creas nuevos artefactos, son las siguientes:

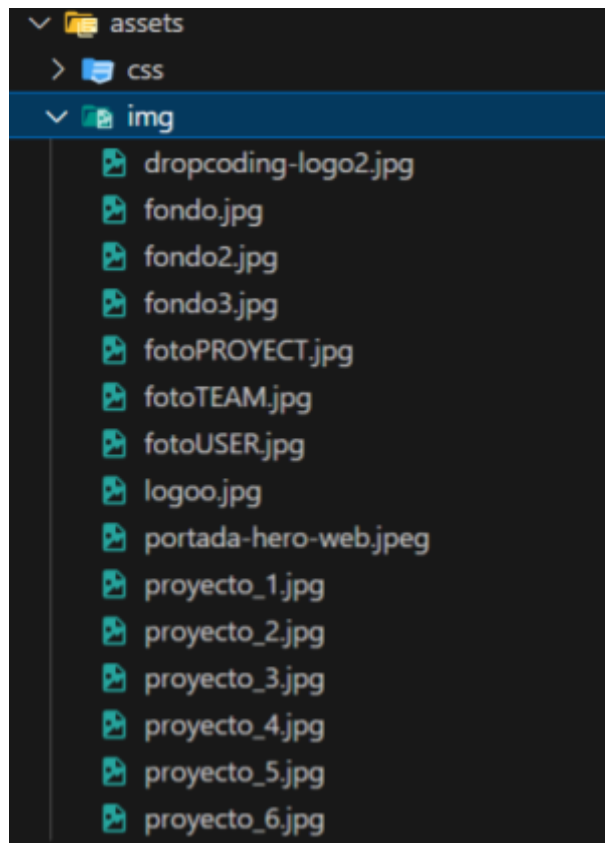


Ilustración 25: Contenido carpeta CSS

- Carpeta “**js**”: carpeta que contiene las diferentes clases .js que definen objetos y crean instancias de esos objetos en cada pantalla de la aplicación, son las siguientes:

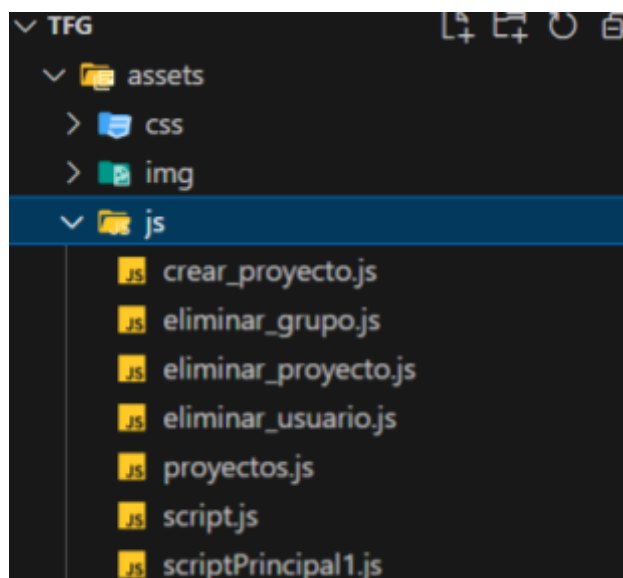


Ilustración 26: Contenido carpeta IMG

Clase	Descripción
crear_proyecto.js	Clase que configura la interacción con los botones y enlaces de la pantalla de creación de un proyecto.
eliminar_grupo.js	Clase que configura la interacción con los botones y enlaces de la pantalla de eliminación de un grupo.
eliminar_proyecto.js	Clase que configura la interacción con los botones y enlaces de la pantalla de eliminación de un proyecto.
eliminar_usuario.js	Clase que configura la interacción con los botones y enlaces de la pantalla de eliminación de un usuario.
proyectos.js	Clase que configura la interacción con los botones y enlaces de la pantalla de visualización de proyectos.
script.js	Clase que configura la interacción con los botones y enlaces de la pantalla de registro e inicio de sesión.
script_Principal1.js	Clase que configura la interacción con los botones y enlaces de la pantalla de bienvenida.

- Carpeta “**php**”: dentro de la carpeta php diferenciamos los archivos controladores de la aplicación, son los siguientes:

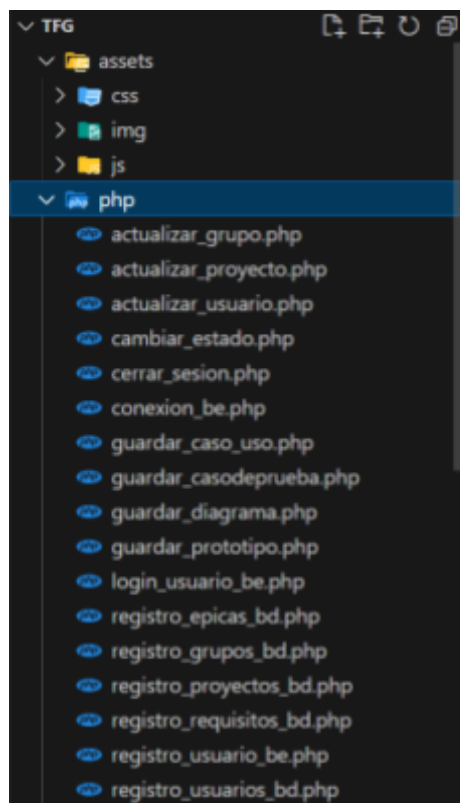


Ilustración 27: Contenido carpeta JS

Clase	Descripción
actualizar_grupo.php	Clase controladora que trabaja como intermediaria con la base de

	datos recibiendo los nuevos datos y actualizando el grupo en ella.
actualizar_proyecto.php	Clase controladora que trabaja como intermediaria con la base de datos recibiendo los nuevos datos y actualizando el proyecto en ella.
actualizar_usuario.php	Clase controladora que trabaja como intermediaria con la base de datos recibiendo los nuevos datos y actualizando un usuario en ella.
cambiar_estado.php	Clase controladora que trabaja como intermediaria con la base de datos recibiendo el dato del estado y actualizando el nuevo estado en ella.
cerrar_sesion.php	Clase controladora que maneja que un usuario cierre la sesión y vuelva la página de inicio.
conexion_be.php	Clase controladora que trabaja como intermediaria con la base de datos recibiendo los nuevos datos y actualizando el nuevo grupo en ella.
guardar_caso_uso.php	Clase controladora que establece la conexión con la base de datos.
guardar_casodeprueba.php	Clase controladora que trabaja como intermediaria con la base de datos recibiendo los nuevos datos y guarda el nuevo caso de prueba en ella.
guardar_diagrama.php	Clase controladora que trabaja como intermediaria con la base de datos recibiendo los nuevos datos y guarda el nuevo diagrama en ella.
guardar_prototipo.php	Clase controladora que trabaja como intermediaria con la base de datos recibiendo los nuevos datos y guarda el nuevo prototipo en ella.
login_usuario_be.php	Clase controladora que maneja el inicio de sesión de un usuario en la aplicación, comprobando si ese usuario existe.
registro_epicas_bd.php	Clase controladora que trabaja como intermediaria con la base de datos recibiendo los nuevos datos y registra una nueva épica en ella en ella.
registro_grupos_bd.php	Clase controladora que trabaja como intermediaria con la base de datos recibiendo los nuevos datos y registra un nuevo grupo en ella.
registro_proyectos_bd.php	Clase controladora que trabaja como intermediaria con la base de datos recibiendo los nuevos datos y registra un nuevo proyecto en ella.
registro_requisitos_bd.php	Clase controladora que trabaja como intermediaria con la base de datos recibiendo los nuevos datos y registra un nuevo requisito en ella.
registro_usuario_be.php	Clase controladora que trabaja como intermediaria con la base de

registro_usuarios_bd.php

datos recibiendo los nuevos datos y registra un nuevo administrador en ella.

Clase controladora que trabaja como intermediaria con la base de datos recibiendo los nuevos datos y registra un nuevo usuario/trabajador en ella.

- Carpeta para el **Backend/Lógica del Servidor**: clases PHP para el manejo de datos interno de toda la aplicación.

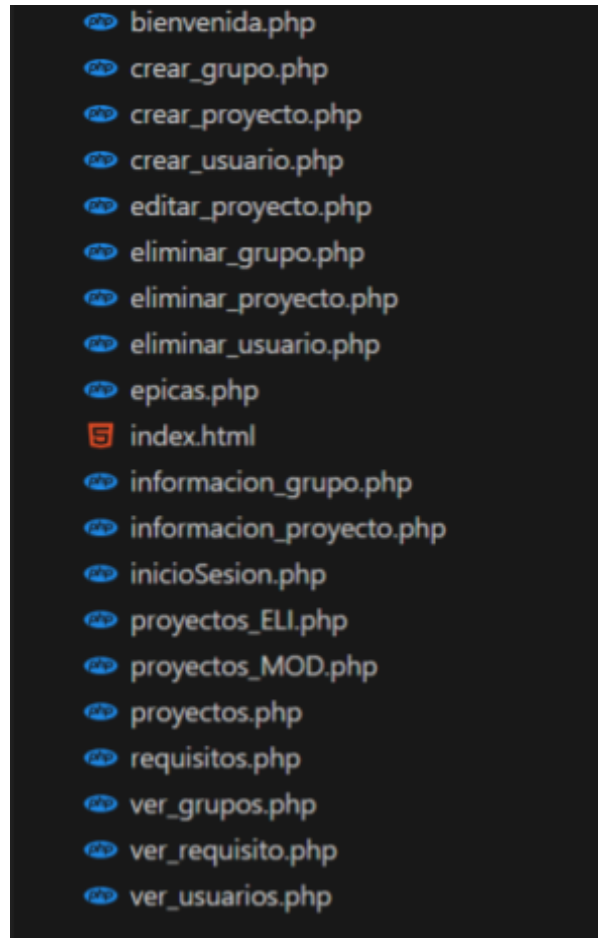


Ilustración 28: Contenido para el backend

Clase	Descripción
bienvenida.php	Clase php para la interfaz de la pantalla de bienvenida de la aplicación.
crear_grupo.php	Clase php para la interfaz de la pantalla de creación de grupos de la aplicación.
crear_proyecto.php	Clase php para la interfaz de la pantalla de creación de proyectos de la aplicación.

crear_usuario.php	Clase php para la interfaz de la pantalla de creación de usuarios de la aplicación.
editar_proyecto.php	Clase php para la interfaz de la pantalla de edición de proyectos de la aplicación.
eliminar_grupo.php	Clase php para la interfaz de la pantalla de eliminación de grupo de la aplicación.
eliminar_proyecto.php	Clase php para la interfaz de la pantalla de eliminación de proyecto de la aplicación.
eliminar_usuario.php	Clase php para la interfaz de la pantalla de bienvenida de la aplicación.
epicas.php	Clase php para la interfaz de la pantalla de creación y listado de épicas de la aplicación.
index.html	Clase html para la interfaz de la pantalla de inicio al entrar en la aplicación antes de iniciar sesión.
informacion_grupo.php	Clase php para la interfaz de la pantalla de información de grupo de la aplicación.
informacion_proyecto.php	Clase php para la interfaz de la pantalla de información de proyecto de la aplicación.
inicioSesion.php	Clase php para la interfaz de la pantalla de inicio de sesión y registro de la aplicación.
proyectos_ELI.php	Clase php para la interfaz de la pantalla de eliminación de proyectos de la aplicación.
proyectos_MOD.php	Clase php para la interfaz de la pantalla de modificación de proyectos de la aplicación.
proyectos.php	Clase php para la interfaz de la pantalla de listado de proyectos de la aplicación.
requisitos.php	Clase php para la interfaz de la pantalla de listado de requisitos de la aplicación.
ver_grupos.php	Clase php para la interfaz de la pantalla de visualización de grupos de la aplicación.
ver_requisitos.php	Clase php para la interfaz de la pantalla de visualización de requisitos de la aplicación.
ver_usuarios.php	Clase php para la interfaz de la pantalla de visualización de usuarios de la aplicación.

6.3. Detalles de la implementación

En esta parte se mostrarán fragmentos importantes del código explicando su funcionamiento los cuales son cruciales para la aplicación, son los siguientes:

6.3.1. Programación de la conexión con la base de datos

- Clase **“conexion_be.php”**: aquí podemos ver la conexión con la base de datos.

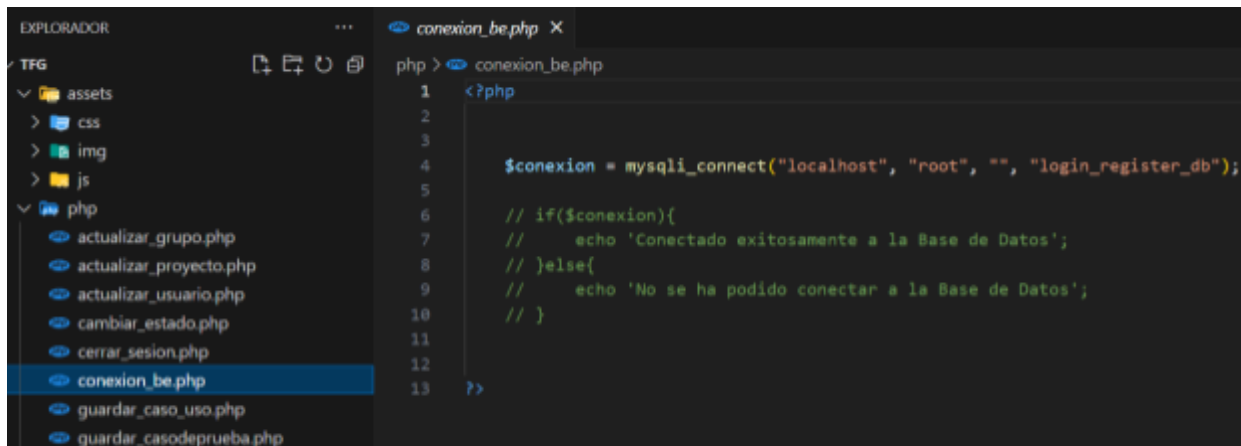


Ilustración 29: Clase “conexión_be.php”

6.3.2. Programación de la pantalla de registro e inicio de sesión

- Clase **“inicioSesion.php”**: en esta clase se toman los datos que se pasarán a las clases **“login_usuario_be.php”** y **“registro_usuario_be.php”** respectivamente, dependiendo si se está registrando o iniciando sesión.

```

inicioSesion.php
1  <?php
2      session_start();
3
4      if(isset($_SESSION['usuario'])){
5          header("location: bienvenida.php");
6      }
7
8  ?>
9  <!DOCTYPE html>
10 <html lang="en">
11 <head>
12     <meta charset="UTF-8">
13     <meta name="viewport" content="width=device-width, initial-scale=1.0">
14     <title>Login y Register</title>
15
16     <link href="https://fonts.googleapis.com/css2?family=Roboto:ital,wght@0,100;0,300;0,400;0,500;0,700;0,900;1,100;1,300;1,400;1,500;1,700;1,900&display=block">
17
18
19     <link rel="stylesheet" href="assets/css/styleLogin.css">
20 </head>
21 <body>
22
23     <main>
24
25         <div class="contenedor_todo">
26             <div class="caja_trasera">
27                 <div class="caja_trasera-login">
28                     <h3>¿Ya tienes una cuenta?</h3>
29                     <p>Inicia sesión para entrar en la página</p>
30                     <button id="btn_iniciar-sesion">Iniciar Sesión</button>
31
32                 </div>
33                 <div class="caja_trasera-register">
34                     <h3>¿Aún no tienes una cuenta?</h3>
35                     <p>Regístrate para que puedas iniciar sesión</p>
36                     <button id="btn_registrarse">Regístrarse</button>
37                 </div>
38             </div>
39
40             <!--Formulario de Login y registro-->
41             <div class="contenedor_login-register">
42                 <!--Login-->
43                 <form action="php/login_usuario_be.php" method="POST" class="formulario_login">
44                     <h2>Iniciar Sesión</h2>
45                     <input type="text" placeholder="Correo Electronico" name="correo">
46                     <input type="password" placeholder="Contraseña" name="contrasena">
47                     <button>Entrar</button>
48                 </form>
49
50                 <!--Register-->
51                 <form action="php/registro_usuario_be.php" method="POST" class="formulario_registro">
52                     <h2>Regístrarse</h2>
53                     <input type="text" placeholder="Nombre completo" name="nombre_completo">
54                     <input type="text" placeholder="Correo Electronico" name="correo">

```

```

54         <input type="text" placeholder="Usuario" name="usuario">
55         <input type="password" placeholder="Contraseña" name="contrasena">
56         <button>Registrarse</button>
57     </form>
58 </div>
59 </div>
60
61 </main>
62
63 <script src="assets/js/script.js"></script>
64 </body>
65 </html>

```

Ilustración 30: Clase "inicioSesion.php"

- Clase **"login_usuario_be.php"**: Aquí recibimos los datos del usuario registrado y mediante una consulta, para comprobar si el usuario y la contraseña es correcta(están registrados previamente en la base de datos), entramos en la aplicación registrados.

```

login_usuario_be.php X
php > login_usuario_be.php
1  <?php
2
3      session_start();
4      include 'conexion_be.php';
5
6      $correo = $_POST['correo'];
7      $contrasena = $_POST['contrasena'];
8      $contrasena = hash('sha512', $contrasena);
9
10
11     $validar_login = mysqli_query($conexion, "SELECT * FROM usuarios WHERE correo='$correo'
12     and contrasena='$contrasena'");
13
14     if(mysqli_num_rows($validar_login) > 0){
15         $_SESSION['usuario'] = $correo;
16         header("location: ../bienvenida.php");
17         exit();
18     }else{
19         echo '
20         <script>
21             alert("El usuario no existe, por favor verifique los datos introducidos");
22             window.location = "../inicioSesion.php";
23         </script>
24         ';
25         exit();
26     }
27 >>

```

Ilustración 31: Clase "login_usuario_be.php"

- Clase “registro_usuario_be.php”: Aquí recibimos los datos del usuario registrado y, mediante un INSERT INTO, registramos al nuevo administrador en la base de datos.

```
registro_usuario_be.php X
php > registro_usuario_be.php
1  <?php
2
3      include 'conexion_be.php';
4
5      $nombre_completo = $_POST['nombre_completo'];
6      $correo = $_POST['correo'];
7      $usuario = $_POST['usuario'];
8      $contrasena = $_POST['contrasena'];
9      //encriptamiento contrasena
10     $contrasena = hash('sha512', $contrasena);
11
12     $query = "INSERT INTO usuarios(nombre_completo, correo, usuario, contrasena)
13             VALUES('$nombre_completo', '$correo', '$usuario', '$contrasena')";
14
15     //verificar que el correo no se repita en la base de datos
16     $verificar_correo = mysqli_query($conexion, "SELECT * FROM usuarios WHERE correo='$correo' ");
17
18     if(mysqli_num_rows($verificar_correo) > 0){
19         echo '
20             <script>
21                 alert("Este correo ya está registrado, pruebe con otro diferente");
22                 window.location = "../inicioSesion.php";
23             </script>
24         ';
25         exit();
26     }
27
28     //verificar que el usuario no se repita en la base de datos
29     $verificar_usuario = mysqli_query($conexion, "SELECT * FROM usuarios WHERE usuario='$usuario' ");
30
31     if(mysqli_num_rows($verificar_usuario) > 0){
32         echo '
33             <script>
34                 alert("Este usuario ya está registrado, pruebe con otro diferente");
35                 window.location = "../inicioSesion.php";
36             </script>
37         ';
38         exit();
39     }
40
41     $ejecutar = mysqli_query($conexion, $query);
42
43     if($ejecutar){
44         echo '
45             <script>
46                 alert("Usuario almacenado exitosamente");
47                 window.location = "../inicioSesion.php";
48             </script>
49         ';
50     }else{
51         echo '
52             <script>
53                 alert("Inténtalo de nuevo, usuario almacenado");
54                 window.location = "../inicioSesion.php";
55             </script>
56         ';
57     }
58     mysqli_close($conexion);
59     ?>
```

Ilustración 32: Clase “registro_usuario_be.php”

6.3.3. Programación del menú lateral

En este fragmento de código programamos el menú lateral izquierdo, el cual mediante `href=""` vamos mandando al administrador a cada una de las secciones de la aplicación.

```
73 <div class="menu__side" id="menu_side">
74
75     <div class="name__page">
76         
77         <h4>Navarro00 RMT</h4>
78     </div>
79
80     <div class="options__menu">
81
82         <a href="bienvenida.php" class="selected">
83             <div class="option">
84                 <i class="fas fa-home" title="Inicio"></i>
85                 <h4>Inicio</h4>
86             </div>
87         </a>
88
89         <a href="proyectos.php">
90             <div class="option">
91                 <i class="far fa-file" title="Portafolio"></i>
92                 <h4>Proyectos</h4>
93             </div>
94         </a>
95
96         <a href="#" id="blog_link">
97             <div class="option">
98                 <i class="far fa-sticky-note" title="Blog"></i>
99                 <h4>Administración</h4>
100             </div>
```

```

101     </a>
102
103     <div id="extra_links" style="display: none;">
104         <a href="crear_proyecto.php">
105             <div class="option" style="padding-left: 30px;">
106                 <i class="fa-solid fa-plus" title="Extra Link 1"></i>
107                 <h4>Crear proyecto</h4>
108             </div>
109         </a>
110         <a href="proyectos_MOD.php">
111             <div class="option" style="padding-left: 30px;">
112                 <i class="fa-solid fa-pen-to-square" title="Extra Link 2"></i>
113                 <h4>Modificar proyecto</h4>
114             </div>
115         </a>
116         <a href="proyectos_ELI.php">
117             <div class="option" style="padding-left: 30px;">
118                 <i class="fa-solid fa-trash" title="Extra Link 3"></i>
119                 <h4>Eliminar proyecto</h4>
120             </div>
121         </a>
122     </div>
123
124     <a href="#" id="blog_link2">
125         <div class="option">
126             <i class="fa-solid fa-user" title="Blog"></i>
127             <h4>Usuarios</h4>
128         </div>
129     </a>
130
131     <div id="extra_links2" style="display: none;">
132         <a href="crear_usuario.php">
133             <div class="option" style="padding-left: 30px;">
134                 <i class="fa-solid fa-plus" title="Extra Link 1"></i>
135                 <h4>Crear usuario</h4>
136             </div>
137         </a>
138         <a href="ver_usuarios.php">
139             <div class="option" style="padding-left: 30px;">
140                 <i class="fa-solid fa-eye" title="Extra Link 2"></i>
141                 <h4>Ver usuarios</h4>
142             </div>
143         </a>
144     </div>
145
146     <a href="#" id="blog_link3">
147         <div class="option">
148             <i class="fa-solid fa-people-group" title="Blog"></i>
149             <h4>Grupos</h4>

```

```
150         </div>
151     </a>
152
153     <div id="extra_links3" style="display: none;">
154         <a href="crear_grupo.php">
155             <div class="option" style="padding-left: 30px;">
156                 <i class="fa-solid fa-plus" title="Extra Link 3"></i>
157                 <h4>Crear grupo</h4>
158             </div>
159         </a>
160         <a href="ver_grupos.php">
161             <div class="option" style="padding-left: 30px;">
162                 <i class="fa-solid fa-eye" title="Extra Link 3"></i>
163                 <h4>Ver grupos</h4>
164             </div>
165         </a>
166     </div>
167
168     <a href="#">
169         <div class="option">
170             <i class="far fa-id-badge" title="Contacto"></i>
171             <h4>Contacto</h4>
172         </div>
173     </a>
174 </div>
175 </div>
```

Ilustración 33: Programación menu lateral

6.3.4. Programación del cuadro de usuario registrado

En este fragmento de código programamos el cuadro que muestra el administrador que inició sesión en la parte superior izquierda, primeramente mediante una petición SELECT para comprobar los datos del usuario y finalmente mostrándolos en el recuadro.

```
<?php

session_start();
include 'php/conexion_be.php';

if(!isset($_SESSION['usuario'])){
    echo '
        <script>
            alert("Porfavor, debes de iniciar sesión");
            window.location = "inicioSesion.php";
        </script>
    ';
    session_destroy();
    die();
}

$user = $_SESSION['usuario'];
$sql = "SELECT nombre_completo, correo, usuario, contrasena FROM usuarios WHERE correo='$user'";
$resultado = $conexion->query($sql);

while($data=$resultado->fetch_assoc()){
    $nombre_completo = $data['nombre_completo'];
    $correo = $data['correo'];
    $usuario = $data['usuario'];
    $contrasena = $data['contrasena'];
}

?>
```

```
69     <header>
70         <div class="icon__menu">
71             <i class="fas fa-bars" id="btn_open"></i>
72         </div>
73
74         <div class="user-info" id="user-info">
75             <div class="user-data">
76                 <p><?php echo $nombre_completo; ?></p>
77                 <p><?php echo $usuario; ?></p>
78             </div>
79         </div>
80
81     </header>
```

Ilustración 34: Programación cuadro de usuario resgistrado

6.3.5. Programación de la pantalla de creación de un proyecto

- Clase “**crear_proyecto.php**”: en esta clase se toman los datos del nuevo proyecto a crear que se pasarán a la clase “**registro_proyectos_bd.php**”.

```

176  <body id="body">
177      <main style="position: relative;">
178          <h1 class="titulo">Crear Proyecto</h1><br>
179          <form method="POST" enctype="multipart/form-data">
180              <!-- NOMBRE -->
181              <label for="project_name" class="label-project-name">Nombre del proyecto:</label>
182              <input type="text" placeholder="Escribe aquí" name="nombre" id="project_name" class="input-project-name">
183              <!-- DESCRIPCION -->
184              <label for="descripcion" class="label-descripcion">Descripción:</label>
185              <input type="text" placeholder="Escribe aquí" name="descripcion" id="descripcion" class="input-descripcion">
186              <!-- METODOLOGIA -->
187              <label for="metodologia" class="label-metodologia">Metodología:</label>
188              <input type="radio" id="agil_checkbox" name="agil" class="person-checkbox">
189              <label for="agil_checkbox" class="person-label">Ágil</label>
190              <br>
191              <input type="radio" id="tradicional_checkbox" name="tradicional" class="person-checkbox2">
192              <label for="tradicional_checkbox" class="person-label2">Tradicional</label>
193              <!-- tipo de requisito -->
194              <h2 class="tipo_requisito">Tipos de requisitos disponibles:</h2>
195              <input type="checkbox" id="option1_checkbox" name="option1" class="option-checkbox">
196              <label for="option1_checkbox" class="option-label3" >Funcionales</label>
197              <br>
198              <input type="checkbox" id="option2_checkbox" name="option2" class="option-checkbox2">
199              <label for="option2_checkbox" class="option-label4" >No Funcionales</label>
200              <br>
201              <input type="checkbox" id="option3_checkbox" name="option3" class="option-checkbox3">
202              <label for="option3_checkbox" class="option-label5" >De Interfaz</label>
203              <br>
204              <input type="checkbox" id="option4_checkbox" name="option4" class="option-checkbox4" >
205              <label for="option4_checkbox" class="option-label6" >De Datos</label>
206              <br>
207              <input type="checkbox" id="option5_checkbox" name="option5" class="option-checkbox5">
208              <label for="option5_checkbox" class="option-label7" >De Mantenimiento</label>
209              <!-- ADJUNTAR FOTO DEL PROYECTO -->
210              <h2 class="foto_portada">Adjuntar foto portada:</h2>
211              <input type="file" id="foto" name="foto" class="photo-input">
212              <!-- BOTON CREAR -->
213              <input type="submit" name="crear" id="create_button" class="create-button">
214          </form>
215          <?php
216              include("php/registro_proyectos_bd.php");
217          ?>
218      </main>

```

Ilustración 35: Clase “crear_proyecto.php”

- Clase “registro_proyectos_bd.php”: Aquí, al igual que en la clase “registro_usuario_be.php”, recibimos los datos del proyecto a crear y, mediante un INSERT INTO, registramos al nuevo proyecto en la base de datos.

```
crear_proyecto.php registro_proyectos_bd.php X
php > registro_proyectos_bd.php
1  <?php
2      //session_start();
3      include 'conexion_be.php';
4
5      if(isset($_POST['crear'])){
6          $nombre = $_POST['nombre'];
7          $descripcion = $_POST['descripcion'];
8          $nombre_imagen = $_FILES['foto']['name'];
9          $temporal = $_FILES['foto']['tmp_name'];
10         $carpeta = 'assets/img';
11         $ruta = $carpeta.'/'.$nombre_imagen;
12         move_uploaded_file($temporal,$carpeta.'/'.$nombre_imagen);
13         if(!isset($_POST['agil']) && !isset($_POST['tradicional'])){
14             echo '
15                 <script>
16                     alert("No has seleccionado metodologia, pruebe otra vez");
17                     window.location = "crear_proyecto.php";
18                 </script>
19             ';
20             exit();
21         }
22         if($nombre == "" || $descripcion == ""){
23             echo '
24                 <script>
25                     alert("Has dejado un campo sin completar, pruebe otra vez");
26                     window.location = "crear_proyecto.php";
27                 </script>
28             ';
29             exit();
30         }

```

```
31     if(isset($_POST['agil'])){
32         $metodologia = "agil";
33     }else{
34         if(isset($_POST['tradicional'])){
35             $metodologia = "tradicional";
36         }
37     }
38
39     $user = $_SESSION['usuario'];
40
41     $query = "INSERT INTO proyectos_bd(usuario, nombre, descripcion, metodologia, ruta)
42         VALUES('$user', '$nombre', '$descripcion', '$metodologia', '$ruta')";
43
44
45     $ejecutar = mysqli_query($conexion, $query);
46
47     if($ejecutar){
48         echo '
49             <script>
50                 alert("Proyecto almacenado exitosamente");
51             </script>
52         ';
53     }
54 }
```

Ilustración 36: Clase "registro_proyectos_bd.php"

6.3.6. Programación de la pantalla de creación y listado de requisitos.

- Clase "requisitos.php":
 - Primeramente mostramos el panel de enlaces que nos llevó hasta donde estamos.
 - Posteriormente mostramos el botón que nos desplegará el formulario de creación de un nuevo requisito y, al igual que en la clase "registro_usuario_be.php", recibimos los datos del requisito a crear y, mediante un INSERT INTO en la clase "registro_requisitos_bd.php", registramos al nuevo requisito en la base de datos.
 - Finalmente mostramos el listado de todos los requisitos ordenamos por estado y con sus respectivas épicas.

```

requisitos.php X
requisitos.php
63 <body id="body">
196
197 <main>
198 <!-- enlaces arriba -->
199 <div class="breadcrumbs">
200 <a href="proyectos.php" class="breadcrumb-link">Proyectos</a>
201 <span class="breadcrumb-symbol">></span>
202 <a href="informacion_proyecto.php?id=<?php echo $info_proyecto['id']; ?>" class="breadcrumb-link">Proyecto <?php echo $info_pro
203 <span class="breadcrumb-symbol">></span>
204 <a class="breadcrumb-link">Requisitos</a>
205 </div>
206
207 <div class="centered-content">
208 <a href="#" class="add-epic-button" id="add_epic_button">
209 <div class="aniade">
210 <i class="fa-solid fa-plus" title="Extra Link 1"></i>
211 <h4>Añadir Requisito</h4>
212 </div>
213 </a>
214
215 <!-- Formulario para añadir épica, inicialmente oculto -->
216 <form action="php/registro_requisitos_bd.php?id=<?php echo $proyecto_id; ?>" id="add_epic_form" style="display: none; margin-to
217 <div class="form-group">
218 <label for="nombre" class="form-label"><strong>Título del requisito:</strong></label>
219 <input type="text" id="nombre" name="nombre" required class="form-input-sm">
220 </div>
221 <div class="form-group">
222 <label for="descripcion" class="form-label"><strong>Descripción:</strong></label>
223 <textarea id="descripcion" name="descripcion" class="form-textarea-sm" placeholder="Escribe aquí" required></textarea>
224 </div>
225
226 <div class="form-group-cajas-container">
227 <div class="form-group-cajas">
228 <label for="estado"><strong>Estado:</strong></label>
229 <div class="select-container">
230 <select id="estado" name="estado">
231 <option value="propuesto">Propuesto</option>
232 <option value="en_revision">En revisión</option>
233 <option value="en_desarrollo">En desarrollo</option>
234 </select>
235 </div>
236 </div>
237
238 <div class="form-group-cajas">
239 <label for="prioridad"><strong>Prioridad:</strong></label>
240 <div class="select-container">
241 <select id="prioridad" name="prioridad">
242 <option value="baja">Baja</option>
243 <option value="media">Media</option>
244 <option value="alta">Alta</option>
245 </select>
246 </div>
247 </div>

```

```

248
249     <div class="form-group-cajas">
250         <label for="complejidad"><strong>Complejidad:</strong></label>
251         <div class="select-container">
252             <select id="complejidad" name="complejidad">
253                 <option value="baja">Baja</option>
254                 <option value="media">Media</option>
255                 <option value="alta">Alta</option>
256             </select>
257         </div>
258     </div>
259
260     <?php
261         $sql_epicas = "SELECT * FROM epicas_bd WHERE proyecto_id = '$proyecto_id'";
262         $resultado_epicas = $conexion->query($sql_epicas);
263
264         // Almacenar los resultados de $resultado_grupos en un array
265         $epicas_array = array();
266         while($epicas = $resultado_epicas->fetch_assoc()) {
267             $epicas_array[] = $epicas;
268         }
269     >>
270
271     <div class="form-group-cajas">
272         <label for="epica"><strong>Selecciona Epica:</strong></label>
273
274         <div class="select-container">
275             <select id="epica" name="epica">
276                 <?php foreach($epicas_array as $epicas) {
277                     $selected = ""; // Define la variable $selected
278                     echo "<option value='" . $epicas['id'] . "' $selected>" . $epicas['nombre'] . "</option>";
279                 } ?>
280             </select>
281         </div>
282     </div>
283
284     <div class="form-group-cajas">
285         <label for="coste"><strong>Coste:</strong></label>
286         <div class="input-container">
287             <input type="number" id="coste" name="coste" step="0.01" pattern="^\d+(\.\d{1,2})?$" placeholder="€">
288         </div>
289     </div>
290
291     <div class="form-group-cajas">
292         <label for="tiempo"><strong>Tiempo:</strong></label>
293         <div class="input-container">
294             <input type="number" id="tiempo" name="tiempo" step="1" pattern="^\d+(\.\d{1,2})?$" placeholder="hrs">
295         </div>
296     </div>

```

```

299         <div class="form-group centered" style="margin-top: 20px;">
300             <input type="submit" name="crear" value="Crear" class="submit-button">
301         </div>
302     </form>
303 </div>
304 <div class="table-container">
305     <table class="modern-table">
306         <thead>
307             <tr>
308                 <th>ÉPICA</th>
309                 <th>REQUISITOS ASOCIADOS</th>
310             </tr>
311         </thead>
312         <tbody>
313             <?php
314                 // Recorrer las epicas y mostrar los requisitos asociados
315                 foreach($epicas_array as $epica) {
316                     echo "<tr>";
317                     echo "<td>" . $epica['nombre'] . "</td>";
318                     echo "<td>";
319
320                     // Consultar requisitos asociados a esta epica
321                     $epica_id = $epica['id'];
322                     //Propuestos
323
324                     //Propuestos
325                     $sql_requisitos1 = "SELECT * FROM requisitos_bd WHERE epica = '$epica_id' AND estado='propuesto'";
326                     $resultado_requisitos1 = $conexion->query($sql_requisitos1);
327
328                     if ($resultado_requisitos1->num_rows > 0) {
329                         echo "<h5>PROPUESTOS</h5>";
330
331                         while($requisito = $resultado_requisitos1->fetch_assoc()) {
332                             echo "<a href='ver_requisito.php?id=" . $requisito['id'] . "'>" . " " . $requisito['nombre'] . "</a><br>";
333                         }
334
335                     // En revisión
336                     $sql_requisitos2 = "SELECT * FROM requisitos_bd WHERE epica = '$epica_id' AND estado='en_revision'";
337                     $resultado_requisitos2 = $conexion->query($sql_requisitos2);
338
339                     if ($resultado_requisitos2->num_rows > 0) {
340                         echo "<h5>EN REVISIÓN</h5>";
341
342                         while($requisito = $resultado_requisitos2->fetch_assoc()) {
343                             echo "<a href='ver_requisito.php?id=" . $requisito['id'] . "'>" . " " . $requisito['nombre'] . "</a><br>";
344                         }
345
346                     // En desarrollo
347                     $sql_requisitos3 = "SELECT * FROM requisitos_bd WHERE epica = '$epica_id' AND estado='en_desarrollo'";
348                     $resultado_requisitos3 = $conexion->query($sql_requisitos3);
349
350                     if ($resultado_requisitos3->num_rows > 0) {
351                         echo "<h5>EN DESARROLLO</h5>";
352
353                         while($requisito = $resultado_requisitos3->fetch_assoc()) {
354                             echo "<a href='ver_requisito.php?id=" . $requisito['id'] . "'>" . " " . $requisito['nombre'] . "</a><br>";
355                         }
356
357                     // Finalizados
358                     $sql_requisitos4 = "SELECT * FROM requisitos_bd WHERE epica = '$epica_id' AND estado='Finalizado'";
359                     $resultado_requisitos4 = $conexion->query($sql_requisitos4);
360
361                     if ($resultado_requisitos4->num_rows > 0) {
362                         echo "<h5>FINALIZADOS</h5>";
363
364                         while($requisito = $resultado_requisitos4->fetch_assoc()) {
365                             echo "<a href='ver_requisito.php?id=" . $requisito['id'] . "'>" . " " . $requisito['nombre'] . "</a><br>";
366                         }
367
368                     }
369
370                     echo "</td>";

```

Ilustración 37: Clase "requisitos.php"

```

requisitos.php  registro_requisitos_bd.php X
php > registro_requisitos_bd.php
1
2 <?php
3     //session_start();
4     include 'conexion_be.php';
5
6     if(isset($_POST['crear'])) {
7         // Obtener los datos del formulario
8         $nombre = $_POST['nombre'];
9         $descripcion = $_POST['descripcion'];
10        $estado = $_POST['estado'];
11        $prioridad = $_POST['prioridad'];
12        $complejidad = $_POST['complejidad'];
13        $epica = $_POST['epica'];
14        $coste = $_POST['coste'];
15        $tiempo = $_POST['tiempo'];
16        $proyecto_id = $_GET['id']; // ID del proyecto obtenido de la URL
17
18        if($nombre == "" || $descripcion == "" || $coste == "" || $tiempo == ""){
19            echo '
20                <script>
21                    alert("Has dejado un campo sin completar, pruebe otra vez");
22                    window.location = "requisitos.php";
23                </script>
24            ';
25            exit();
26        }
27
28        // Preparar la consulta SQL para insertar la nueva épica en la base de datos
29        $sql = "INSERT INTO requisitos_bd (nombre, descripcion, proyecto_id, epica, estado, prioridad, complejidad, coste, tiempo) VALUES (
30
31
32        // Ejecutar la consulta
33        if($conexion->query($sql) === TRUE) {
34            echo '
35                <script>
36                    alert("Requisito creado exitosamente");
37                    window.location = "../requisitos.php?id=" . urlencode($proyecto_id) . " ";
38                </script>
39            ';
40        } else {
41            echo "Error al crear el requisito: " . $conexion->error;
42        }
43    }
44 }
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Ilustración 38: Clase “registro_requisitos.php”

6.3.7. Programación de la trazabilidad de cada uno de los requisitos.

- En la clase “**ver_requisito.php**” programamos en la segunda columna la trazabilidad que aparecerá cada vez que añadamos un nuevo artefacto, se irá actualizando automáticamente.

```

295 <!-- Segunda columna -->
296 <div class="column">
297     <div class="top-half">
298         <h2>Trazabilidad</h2>
299     </div>
300     <div class="bottom-half trazabilidad-column">
301         <?php
302             // proyecto
303             echo '<div>Proyecto: ' . $proyecto['nombre'] . '</div>';
304             echo '<div class="arrow-down"></div>';
305             // epica
306             $sql_epicas = "SELECT * FROM epicas_bd WHERE id = '" . $info_requisito['epica'] . "'";
307             $resultado_epicas = $conexion->query($sql_epicas);
308             $epica = $resultado_epicas->fetch_assoc();
309             echo '<div>Épica: ' . $epica['nombre'] . '</div>';
310             echo '<div class="arrow-down"></div>';
311             // requisito
312             echo '<div>Requisito: ' . $info_requisito['nombre'] . '</div>';
313             // Mostrar la flecha apuntando hacia abajo
314             echo '<div class="arrow-down"></div>';
315             // Obtener y mostrar los casos de uso asociados al requisito
316             $sql_casos_uso = "SELECT * FROM casodeuso_bd WHERE requisito_id = '" . $info_requisito['id'] . "'";
317             $resultado_casos_uso = $conexion->query($sql_casos_uso);
318             if ($resultado_casos_uso->num_rows > 0) {
319                 echo '<div>Casos de uso: </div>';
320                 echo '<ul>';
321
322                 while ($caso_uso = $resultado_casos_uso->fetch_assoc()) {
323                     echo '<li>' . $caso_uso['titulo'] . '</li>';
324                 }
325                 echo '</ul>';
326             } else {
327                 echo '<div>No se encontraron casos de uso asociados.</div>';
328             }
329             // Obtener y mostrar los Prototipos asociados al requisito
330             echo '<div class="arrow-down"></div>';
331             $sql_prototipo = "SELECT * FROM prototipo_bd WHERE requisito_id = '" . $info_requisito['id'] . "'";
332             $resultado_prototipo = $conexion->query($sql_prototipo);
333             if ($resultado_prototipo->num_rows > 0) {
334                 echo '<div>Prototipos de interfaz de usuario: </div>';
335                 echo '<ul>';
336                 while ($prototipo = $resultado_prototipo->fetch_assoc()) {
337                     echo '<li>' . $prototipo['titulo'] . '</li>';
338                 }
339                 echo '</ul>';
340             } else {
341                 echo '<div>No se encontraron prototipos asociados.</div>';
342             }
343             // Obtener y mostrar los diagramas asociados al requisito
344             echo '<div class="arrow-down"></div>';
345             $sql_diagramas = "SELECT * FROM diagrama_bd WHERE requisito_id = '" . $info_requisito['id'] . "'";
346             $resultado_diagrama = $conexion->query($sql_diagramas);

```

```

345 $resultado_diagrama = $conexion->query($sql_diagramas);
346 if ($resultado_diagrama->num_rows > 0) {
347     echo '<div>Diagramas de clases: </div>';
348     echo '<ul>';
349     while ($diagrama = $resultado_diagrama->fetch_assoc()) {
350         echo '<li>' . $diagrama['titulo'] . '</li>';
351     }
352     echo '</ul>';
353 } else {
354     echo '<div>No se encontraron diagramas asociados.</div>';
355 }
356 // Obtener y mostrar los casos de prueba asociados al requisito
357 echo '<div class="arrow-down"></div>';
358 $sql_prueba = "SELECT * FROM paso_paso_bd WHERE requisito_id = '" . $info_requisito['id'] . "'";
359 $resultado_prueba = $conexion->query($sql_prueba);
360 if ($resultado_prueba->num_rows > 0) {
361     echo '<div>Casos de prueba: </div>';
362     echo '<ul>';
363     while ($prueba = $resultado_prueba->fetch_assoc()) {
364         echo '<li>' . $prueba['titulo'] . '</li>';
365     }
366     echo '</ul>';
367 } else {
368     echo '<div>No se encontraron casos de prueba asociados.</div>';
369 }

```

Ilustración 39: Clase "ver_requisito.php"

7. PRUEBAS

Para las pruebas del proyecto se realizará el estudio mediante pruebas de caja negra, las cuales son una técnica de software que se centran en evaluar la funcionalidad del sistema sin conocer su estructura interna o código fuente. En otras palabras, el evaluador no necesita saber cómo funciona internamente el software, solo se preocupa por verificar si el software produce los resultados correctos según los requisitos especificados.

A continuación se mostrarán las pruebas realizadas con los resultados obtenidos.:

7.1. CP-01 Registro de usuario

CP-01 Registro de usuario

Objetivo	Comprobar si es posible registrarse en el sistema.						
Precondiciones							
Acción esperada	1. El usuario puede registrar sus datos en el sistema. 2. Poder pasar a la pantalla de inicio de sesión.						
Secuencia	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>Introducir datos</td></tr> <tr> <td>2</td><td>Pulsar en "Registrarse"</td></tr> </tbody> </table>	Paso	Acción	1	Introducir datos	2	Pulsar en "Registrarse"
Paso	Acción						
1	Introducir datos						
2	Pulsar en "Registrarse"						
Resultado	Correcto						
Requisitos Superados	FRQ-01: Registro en el sistema						

7.2. CP-02 Inicio de sesión

CP-02 Inicio de sesión

Objetivo	Comprobar si es posible iniciar sesión en el sistema.						
Precondiciones	Estar registrado en el sistema						
Acción esperada	1. El usuario puede ingresar sus datos en el sistema. 2. Poder pasar a la pantalla de bienvenida.						
Secuencia	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>Introducir usuario y contraseña</td></tr> <tr> <td>2</td><td>Pulsar en "Entrar"</td></tr> </tbody> </table>	Paso	Acción	1	Introducir usuario y contraseña	2	Pulsar en "Entrar"
Paso	Acción						
1	Introducir usuario y contraseña						
2	Pulsar en "Entrar"						

Resultado

Correcto

Requisitos Superados

FRQ-02: Inicio de sesión

7.3. CP-03 Cerrar sesión

CP-03 Cerrar sesión

Objetivo	Comprobar si es posible cerrar sesión en el sistema.						
Precondiciones	Haber iniciado sesión en el sistema.						
Acción esperada	1. El usuario al pulsar sobre su nombre debe aparecer el botón “cerrar sesión”. 2. Poder pulsar el botón y cerrar sesión en el sistema. 3. Pasar a la pantalla de registro nuevamente.						
Secuencia	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>Pulsar sobre tu nombre de usuario</td></tr> <tr> <td>2</td><td>Pulsar en “Cerrar Sesión”</td></tr> </tbody> </table>	Paso	Acción	1	Pulsar sobre tu nombre de usuario	2	Pulsar en “Cerrar Sesión”
Paso	Acción						
1	Pulsar sobre tu nombre de usuario						
2	Pulsar en “Cerrar Sesión”						
Resultado	Correcto						
Requisitos Superados	FRQ-03: Cierre de sesión						

7.4. CP-04 Menú desplegable

CP-04 Menú desplegable

Objetivo	Comprobar si el menú desplegable funciona.						
Precondiciones	Haber iniciado sesión en el sistema.						
Acción esperada	1. El usuario al pulsar sobre el símbolo de hamburguesa debe aparecer el menú desplegado. 2. Al volver a pulsar debe cerrarse						
Secuencia	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>Pulsar en botón de hamburguesa</td></tr> <tr> <td>2</td><td>Volver a pulsar</td></tr> </tbody> </table>	Paso	Acción	1	Pulsar en botón de hamburguesa	2	Volver a pulsar
Paso	Acción						
1	Pulsar en botón de hamburguesa						
2	Volver a pulsar						

Resultado

Correcto

Requisitos Superados

7.5. CP-05 Crear Proyecto

CP-05 Crear Proyecto

Objetivo	Comprobar si es posible crear un proyecto.										
Precondiciones	Haber iniciado sesión en el sistema.										
Acción esperada	El usuario al pulsar sobre “Administración” debe aparecer el botón “Crear proyecto”. Poder pulsar el botón y que aparezca la pantalla de creación. Poder crear el proyecto.										
Secuencia	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>Pulsar en “Administración”</td></tr> <tr> <td>2</td><td>Pulsar en “Crear proyecto”</td></tr> <tr> <td>3</td><td>Introducir datos</td></tr> <tr> <td>4</td><td>Pulsar en “Enviar”</td></tr> </tbody> </table>	Paso	Acción	1	Pulsar en “Administración”	2	Pulsar en “Crear proyecto”	3	Introducir datos	4	Pulsar en “Enviar”
Paso	Acción										
1	Pulsar en “Administración”										
2	Pulsar en “Crear proyecto”										
3	Introducir datos										
4	Pulsar en “Enviar”										
Resultado	Correcto										
Requisitos Superados	FRQ-14: Crear nuevo proyecto										

7.6. CP-06 Modificar Proyecto

CP-06 Modificar Proyecto

Objetivo	Comprobar si es posible modificar un proyecto.
Precondiciones	Haber iniciado sesión en el sistema. Haber creado un proyecto.
Acción esperada	1. El usuario al pulsar sobre “Administración” debe aparecer el botón “Modificar proyecto”. 2. Poder pulsar el botón y que aparezca la pantalla con los proyectos creados. 3. Pulsar en “Editar” en el proyecto que quieras editar y pasar a la pantalla de edición. 4. Poder editar el proyecto.
Secuencia	

	<table> <tr> <th>Paso</th><th>Acción</th></tr> <tr> <td>1</td><td>Pulsar en “Administración”</td></tr> <tr> <td>2</td><td>Pulsar en “Modificar proyecto”</td></tr> <tr> <td>3</td><td>Pulsar en “Editar”</td></tr> <tr> <td>4</td><td>Introducir nuevos datos</td></tr> <tr> <td>5</td><td>Pulsar en “Actualizar”</td></tr> </table>	Paso	Acción	1	Pulsar en “Administración”	2	Pulsar en “Modificar proyecto”	3	Pulsar en “Editar”	4	Introducir nuevos datos	5	Pulsar en “Actualizar”
Paso	Acción												
1	Pulsar en “Administración”												
2	Pulsar en “Modificar proyecto”												
3	Pulsar en “Editar”												
4	Introducir nuevos datos												
5	Pulsar en “Actualizar”												
Resultado	Correcto												
Requisitos Superados	FRQ-15: Modificar proyecto												

7.7. CP-07 Eliminar Proyecto

CP-07 Eliminar Proyecto

Objetivo	Comprobar si es posible eliminar un proyecto.								
Precondiciones	Haber iniciado sesión en el sistema. Haber creado un proyecto.								
Acción esperada	1. El usuario al pulsar sobre “Administración” debe aparecer el botón “Eliminar proyecto”. 2. Poder pulsar el botón y que aparezca la pantalla con los proyectos creados. 3. Pulsar en “Eliminar ” en el proyecto que quieras eliminados. 4. Poder eliminar el proyecto.								
Secuencia	<table> <tr> <th>Paso</th><th>Acción</th></tr> <tr> <td>1</td><td>Pulsar en “Administración”</td></tr> <tr> <td>2</td><td>Pulsar en “Eliminar proyecto”</td></tr> <tr> <td>3</td><td>Pulsar en “Eliminar”</td></tr> </table>	Paso	Acción	1	Pulsar en “Administración”	2	Pulsar en “Eliminar proyecto”	3	Pulsar en “Eliminar”
Paso	Acción								
1	Pulsar en “Administración”								
2	Pulsar en “Eliminar proyecto”								
3	Pulsar en “Eliminar”								
Resultado	Correcto								
Requisitos Superados	FRQ-16: Eliminar proyecto								

7.8. CP-08 Crear Grupo de trabajo

CP-08 Crear Grupo de trabajo

Objetivo	Comprobar si es posible crear un grupo de trabajo.										
Precondiciones	Haber iniciado sesión en el sistema.										
Acción esperada	1. El usuario al pulsar sobre "Grupos" debe aparecer el botón "Crear grupo". 2. Poder pulsar el botón y que aparezca la pantalla de creación. 3. Poder crear el grupo.										
Secuencia	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>Pulsar en "Grupos"</td></tr> <tr> <td>2</td><td>Pulsar en "Crear grupo"</td></tr> <tr> <td>3</td><td>Introducir datos</td></tr> <tr> <td>4</td><td>Pulsar en "Crear"</td></tr> </tbody> </table>	Paso	Acción	1	Pulsar en "Grupos"	2	Pulsar en "Crear grupo"	3	Introducir datos	4	Pulsar en "Crear"
Paso	Acción										
1	Pulsar en "Grupos"										
2	Pulsar en "Crear grupo"										
3	Introducir datos										
4	Pulsar en "Crear"										
Resultado	Correcto										
Requisitos Superados	FRQ-09: Crear nuevo grupo de trabajo										

7.9. CP-09 Visualizar Grupos de trabajo

CP-09 Visualizar Grupos de trabajo

Objetivo	Comprobar si es posible visualizar los grupos de trabajo.						
Precondiciones	Haber iniciado sesión en el sistema. Haber creado algún grupo.						
Acción esperada	1. El usuario al pulsar sobre "Grupos" debe aparecer el botón "Ver grupos". 2. Poder pulsar el botón y que aparezca la pantalla con el listado.						
Secuencia	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>Pulsar en "Grupos"</td></tr> <tr> <td>2</td><td>Pulsar en "Ver grupos"</td></tr> </tbody> </table>	Paso	Acción	1	Pulsar en "Grupos"	2	Pulsar en "Ver grupos"
Paso	Acción						
1	Pulsar en "Grupos"						
2	Pulsar en "Ver grupos"						
Resultado	Correcto						
Requisitos	FRQ-11: Visualizar grupos de trabajo existentes						

Superados

--

7.10. CP-10 Visualizar Proyectos

CP-10 Visualizar Proyectos

Objetivo	Comprobar si es posible visualizar los proyectos.				
Precondiciones	Haber iniciado sesión en el sistema. Haber creado algún proyecto.				
Acción esperada	1. Poder pulsar el botón “Proyectos” y que aparezca la pantalla con el listado.				
Secuencia	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>Pulsar en “Proyectos”</td></tr> </tbody> </table>	Paso	Acción	1	Pulsar en “Proyectos”
Paso	Acción				
1	Pulsar en “Proyectos”				
Resultado	Correcto				
Requisitos Superados	FRQ-17: Visualizar proyectos existentes				

7.11. CP-11 Crear Usuario

CP-11 Crear Usuario

Objetivo	Comprobar si es posible crear un nuevo usuario.										
Precondiciones	Haber iniciado sesión en el sistema.										
Acción esperada	El usuario al pulsar sobre “Usuarios” debe aparecer el botón “Crear usuario”. Poder pulsar el botón y que aparezca la pantalla de creación. Poder crear el usuario.										
Secuencia	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>Pulsar en “Usuarios”</td></tr> <tr> <td>2</td><td>Pulsar en “Crear usuario”</td></tr> <tr> <td>3</td><td>Introducir datos</td></tr> <tr> <td>4</td><td>Pulsar en “Crear”</td></tr> </tbody> </table>	Paso	Acción	1	Pulsar en “Usuarios”	2	Pulsar en “Crear usuario”	3	Introducir datos	4	Pulsar en “Crear”
Paso	Acción										
1	Pulsar en “Usuarios”										
2	Pulsar en “Crear usuario”										
3	Introducir datos										
4	Pulsar en “Crear”										
Resultado	Correcto										

**Requisitos
Superados**

FRQ-04: Crear nuevo usuario

7.12. CP-12 Visualizar usuarios

CP-12 Visualizar usuarios

Objetivo	Comprobar si es posible visualizar los usuarios.						
Precondiciones	Haber iniciado sesión en el sistema. Haber creado algún usuario.						
Acción esperada	El usuario al pulsar sobre “Usuarios” debe aparecer el botón “Ver usuarios”. Poder pulsar el botón y que aparezca la pantalla con el listado.						
Secuencia	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>Pulsar en “Usuarios”</td></tr> <tr> <td>2</td><td>Pulsar en “Ver usuarios”</td></tr> </tbody> </table>	Paso	Acción	1	Pulsar en “Usuarios”	2	Pulsar en “Ver usuarios”
Paso	Acción						
1	Pulsar en “Usuarios”						
2	Pulsar en “Ver usuarios”						
Resultado	Correcto						
Requisitos Superados							

7.13. CP-13 Crear y visualizar épicas

CP-13 Crear y visualizar épicas

Objetivo	Comprobar si es posible crear y visualizar nuevas épicas.						
Precondiciones	Haber iniciado sesión en el sistema. Haber creado algún proyecto.						
Acción esperada	El usuario al pulsar sobre “Proyectos” debe aparecer el botón “Ver proyecto”. Poder pulsar el botón y que aparezca la pantalla de información del proyecto. El usuario al pulsar sobre “Ir a” del apartado de épicas los llevará a la pantalla de épicas. Poder crear y visualizar las épicas.						
Secuencia	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>Pulsar en “Proyectos”</td></tr> <tr> <td>2</td><td>Pulsar en “Ver proyecto”</td></tr> </tbody> </table>	Paso	Acción	1	Pulsar en “Proyectos”	2	Pulsar en “Ver proyecto”
Paso	Acción						
1	Pulsar en “Proyectos”						
2	Pulsar en “Ver proyecto”						

	<table border="1"> <tr> <td>3</td><td>Pulsar en “Ir a”</td></tr> <tr> <td>4</td><td>Introducir datos</td></tr> <tr> <td>5</td><td>Pulsar en “Crear”</td></tr> </table>	3	Pulsar en “Ir a”	4	Introducir datos	5	Pulsar en “Crear”
3	Pulsar en “Ir a”						
4	Introducir datos						
5	Pulsar en “Crear”						
Resultado	Correcto						
Requisitos Superados	FRQ-19: Crear nueva épica FRQ-20: Visualizar lista de epicas FRQ-21: Ver requisitos asociados a las épicas						

7.14. CP-14 Crear y visualizar requisitos

CP-14 Crear y visualizar requisitos

Objetivo	Comprobar si es posible crear y visualizar nuevos requisitos.												
Precondiciones	Haber iniciado sesión en el sistema. Haber creado algún proyecto. Haber creado alguna épica.												
Acción esperada	El usuario al pulsar sobre “Proyectos” debe aparecer el botón “Ver proyecto”. Poder pulsar el botón y que aparezca la pantalla de información del proyecto. El usuario al pulsar sobre “Ir a” del apartado de requisitos los llevará a la pantalla de requisitos. Poder crear y visualizar los requisitos.												
Secuencia	<table border="1"> <tr> <th>Paso</th><th>Acción</th></tr> <tr> <td>1</td><td>Pulsar en “Proyectos”</td></tr> <tr> <td>2</td><td>Pulsar en “Ver proyecto”</td></tr> <tr> <td>3</td><td>Pulsar en “Ir a”</td></tr> <tr> <td>4</td><td>Introducir datos</td></tr> <tr> <td>5</td><td>Pulsar en “Crear”</td></tr> </table>	Paso	Acción	1	Pulsar en “Proyectos”	2	Pulsar en “Ver proyecto”	3	Pulsar en “Ir a”	4	Introducir datos	5	Pulsar en “Crear”
Paso	Acción												
1	Pulsar en “Proyectos”												
2	Pulsar en “Ver proyecto”												
3	Pulsar en “Ir a”												
4	Introducir datos												
5	Pulsar en “Crear”												
Resultado	Correcto												
Requisitos Superados	FRQ-22: Crear nuevo requisito FRQ-23: Visualizar listado de requisitos												

7.15. CP-15 Ver trazabilidad de requisitos

CP-15 Ver trazabilidad de requisitos

Objetivo	Comprobar si es posible ver la trazabilidad de los requisitos.										
Precondiciones	Haber iniciado sesión en el sistema. Haber creado algún proyecto. Haber creado alguna épica. Haber creado algún requisito.										
Acción esperada	1. El usuario al pulsar algún requisito del listado, nos llevará a la pantalla de información del requisito.										
Secuencia	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> <tr> <td>1</td><td>Pulsar en "Proyectos"</td></tr> <tr> <td>2</td><td>Pulsar en "Ver proyecto"</td></tr> <tr> <td>3</td><td>Pulsar en "Ir a"</td></tr> <tr> <td>5</td><td>Pulsar en algún requisito</td></tr> </tbody> </table>	Paso	Acción	1	Pulsar en "Proyectos"	2	Pulsar en "Ver proyecto"	3	Pulsar en "Ir a"	5	Pulsar en algún requisito
Paso	Acción										
1	Pulsar en "Proyectos"										
2	Pulsar en "Ver proyecto"										
3	Pulsar en "Ir a"										
5	Pulsar en algún requisito										
Resultado	Correcto										
Requisitos Superados	FRQ-24: Ver información del requisito FRQ-25: Cambiar el estado del requisito FRQ-26: Visualizar la trazabilidad del requisito										

7.16. CP-16 Crear nuevos artefactos

CP-16 Crear nuevos artefactos

Objetivo	Comprobar si es posible crear y visualizar nuevos artefactos.		
Precondiciones	Haber iniciado sesión en el sistema. Haber creado algún proyecto. Haber creado alguna épica. Haber creado algún requisito.		
Acción esperada	1. El usuario al pulsar algún requisito del listado, nos llevará a la pantalla de información del requisito. 2. Al pulsar en el desplegable de nuevo artefacto aparecerán 4 opciones con los 4 posibles artefactos a crear. 3. Elegir el que queramos crear y que aparezca un formulario para cada uno. 4. Poder crearlo y que se actualice la trazabilidad del requisito.		
Secuencia	<table border="1"> <thead> <tr> <th>Paso</th><th>Acción</th></tr> </thead> <tbody> </tbody> </table>	Paso	Acción
Paso	Acción		

Resultado**Requisitos
Superados**

	1	Pulsar en "Proyectos"
	2	Pulsar en "Ver proyecto"
	3	Pulsar en "Ir a"
	4	Pulsar en algún requisito
	5	Elegir algún artefacto a crear
	5	Introducir datos
	6	Pulsar en "Crear"
Correcto		
FRQ-27: Añadir caso de uso al requisito FRQ-28: Añadir diagrama de clase al requisito FRQ-29: Añadir prototipo de interfaz de usuario al requisito FRQ-30: Añadir caso de prueba al requisito		

8. MANUALES

8.1. Manual de instalación

Requisitos del Sistema

Antes de proceder con la instalación de la herramienta para la trazabilidad de requisitos en proyectos software, asegúrese de que su sistema cumpla con los siguientes requisitos:

- Sistema Operativo: Windows 7/8/10/11, macOS, o Linux
- Memoria RAM: Mínimo 4 GB
- Espacio en Disco: Mínimo 500 MB disponibles
- Software Requerido:
- XAMPP (versión mínima: 7.4.0)
- Visual Studio Code
- Navegador web moderno (Chrome, Firefox, Edge, etc.)

Pasos de Instalación

1. Instalación de **XAMPP**

- Descargar XAMPP:
 - Visite el sitio oficial de XAMPP.
 - Descargue el instalador adecuado para su sistema operativo.
- Instalar XAMPP:
 - Ejecute el instalador descargado.
 - Siga las instrucciones del asistente de instalación.
 - Asegúrese de seleccionar los componentes Apache, MySQL y PHP.
- Configurar XAMPP:
 - Inicie el Panel de Control de XAMPP.
 - Inicie los servicios de Apache y MySQL.

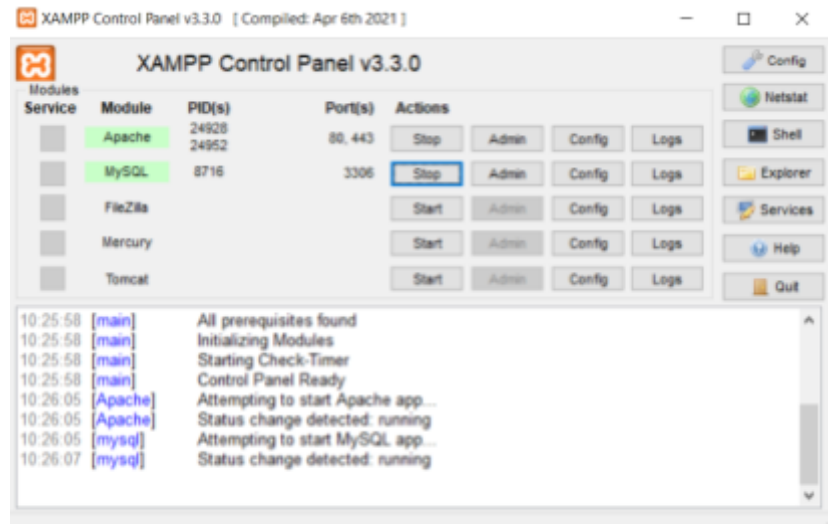


Ilustración 40: XAMPP iniciado

2. Configuración de la Base de Datos

- Crear la Base de Datos:
 - Abra el navegador web y acceda a <http://localhost/phpmyadmin/>.
 - Cree una nueva base de datos, por ejemplo, `gestion_requisitos`.

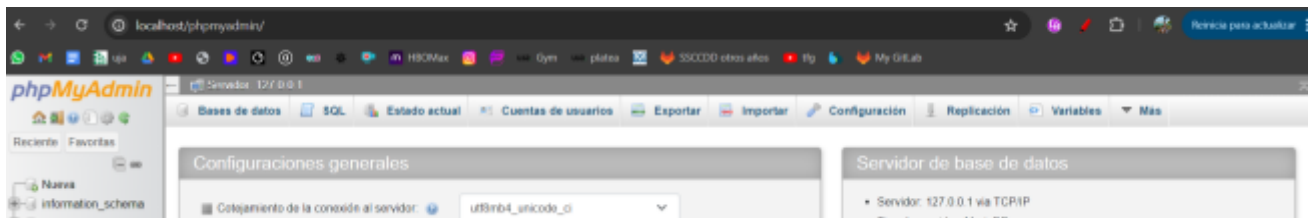


Ilustración 41: Base de datos phpMyAdmin

- Importar la Estructura de la Base de Datos:
 - En phpMyAdmin, seleccione la base de datos creada.
 - Haga clic en la pestaña Importar y cargue el archivo `gestion_requisitos.sql` proporcionado en el paquete de la aplicación.
 - Ejecute la importación para crear las tablas necesarias.

3. Configuración del Entorno de Desarrollo

- Instalar **Visual Studio Code**:
 - Visite el sitio oficial de Visual Studio Code.
 - Descargue e instale Visual Studio Code para su sistema operativo.
- Clonar el Repositorio del Proyecto:
 - Utilice Git para clonar el repositorio del proyecto en su máquina local.
 - En la terminal de comandos, ejecute: `git clone <URL_del_repositorio>`
 - Navegue al directorio del proyecto: `cd <nombre_del_directorio_del_proyecto>`
- Configurar el Archivo de Configuración:
 - Dentro del directorio del proyecto, localice el archivo `config.php`.
 - Edite el archivo `config.php` para establecer los parámetros de conexión a la base de datos:

```
<?php
define('DB_SERVER', 'localhost');
define('DB_USERNAME', 'root');
define('DB_PASSWORD', '');
define('DB_DATABASE', 'gestion_requisitos');
?>
```

4. Despliegue y Pruebas Locales

- Mover los Archivos del Proyecto:
 - Copie todos los archivos y carpetas del proyecto a la carpeta `htdocs` de XAMPP:
- Acceder a la Aplicación:
 - Abra su navegador web y acceda a `http://localhost/<nombre_del_directorio_del_proyecto>`.
 - La página de inicio de sesión debería aparecer, permitiéndole iniciar sesión o registrarse como usuario.

8.2. Manual de usuario

Este manual ha sido creado para guiar a los usuarios a través de las diversas funcionalidades y características de la aplicación, proporcionando una comprensión detallada de cómo utilizarla de manera eficiente y efectiva, mediante ejemplos visuales de uso.

8.2.1. Inicio de la aplicación

Está es la pantalla inicial que se mostrará justo al abrir la dirección web. Para avanzar a la pantalla de registro daremos clic en **“Iniciar Sesión”**.

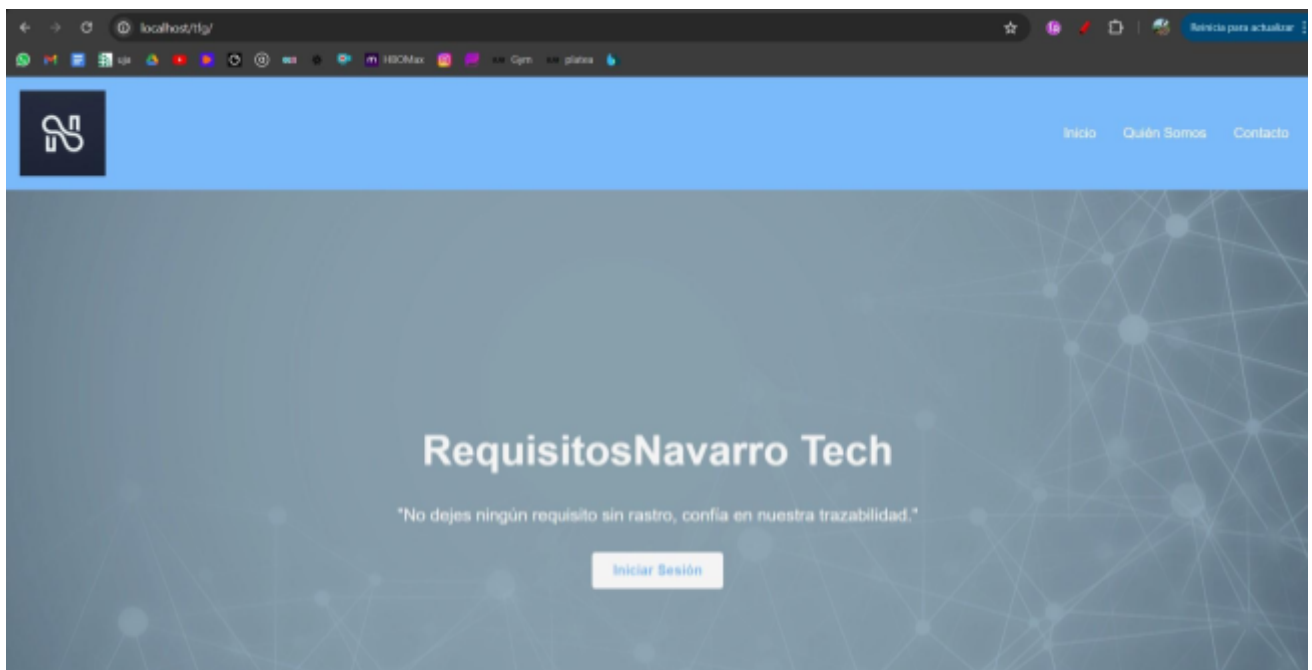


Ilustración 42: Pantalla de inicio de la aplicación

8.2.2. Pantalla de registro e inicio de sesión

Una vez en esta pantalla, si es la primera vez que entras y no tienes un usuario ya creado, lo primero que tienes que hacer es registrarte como se muestra en la imagen, metiendo los datos que nos pide y pulsando el botón **“Registrarse”**.

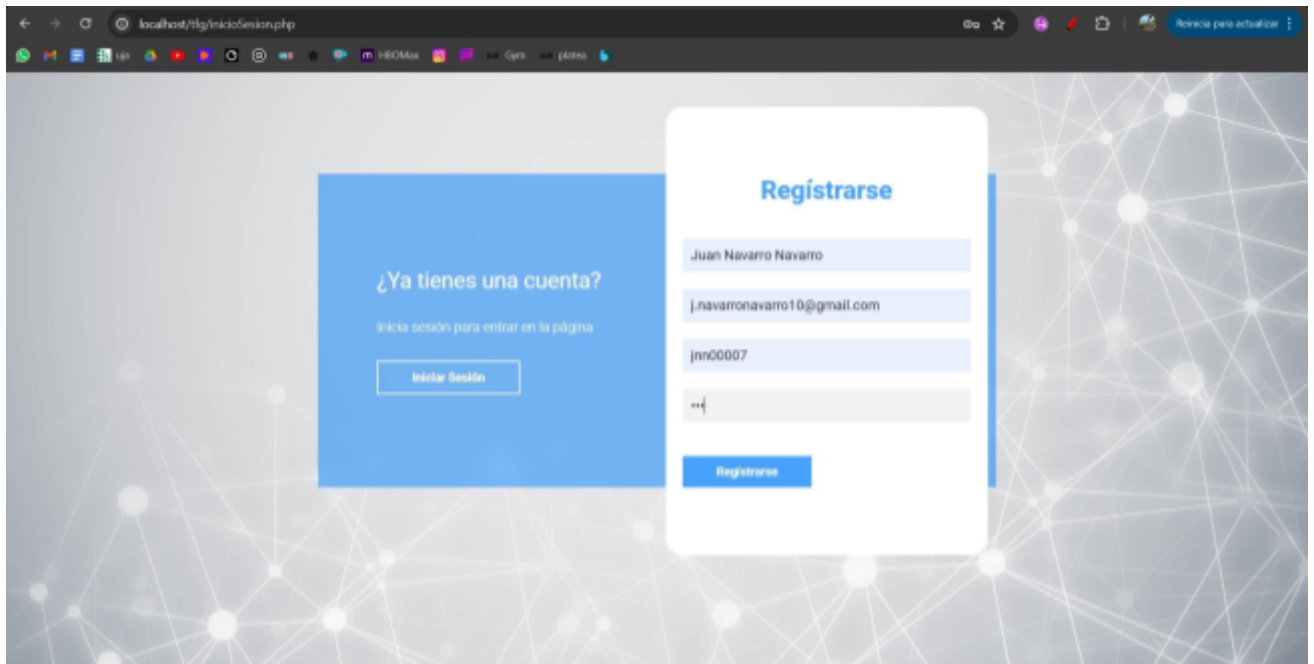


Ilustración 43: Pantalla de registro

Seguidamente pasaremos a la parte de la izquierda para iniciar sesión en el sistema con tu correo electrónico y contraseña anteriormente establecida en el registro, y pulsamos en **“Entrar”**.

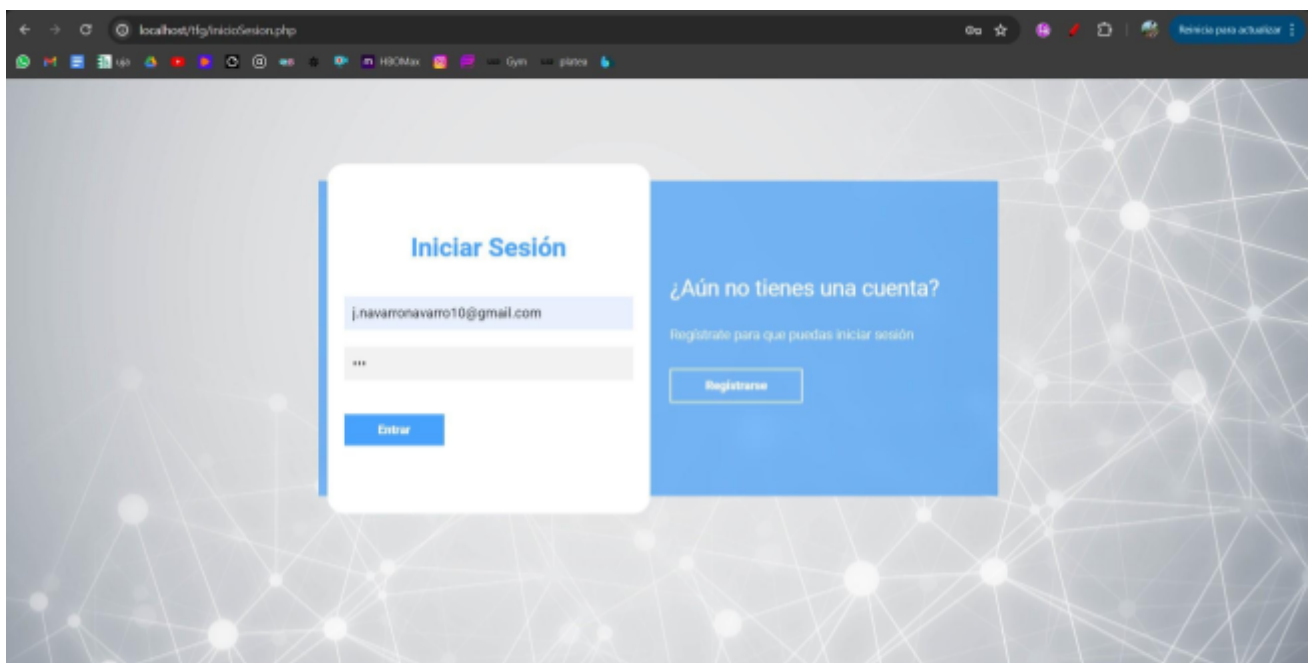


Ilustración 44: Pantalla de inicio de sesión

8.2.3. Pantalla de bienvenida.

Esta pantalla simplemente es la pantalla inicial de la aplicación, en ella encontraremos información, el menú principal y la opción de cerrar sesión que se explicará más adelante.

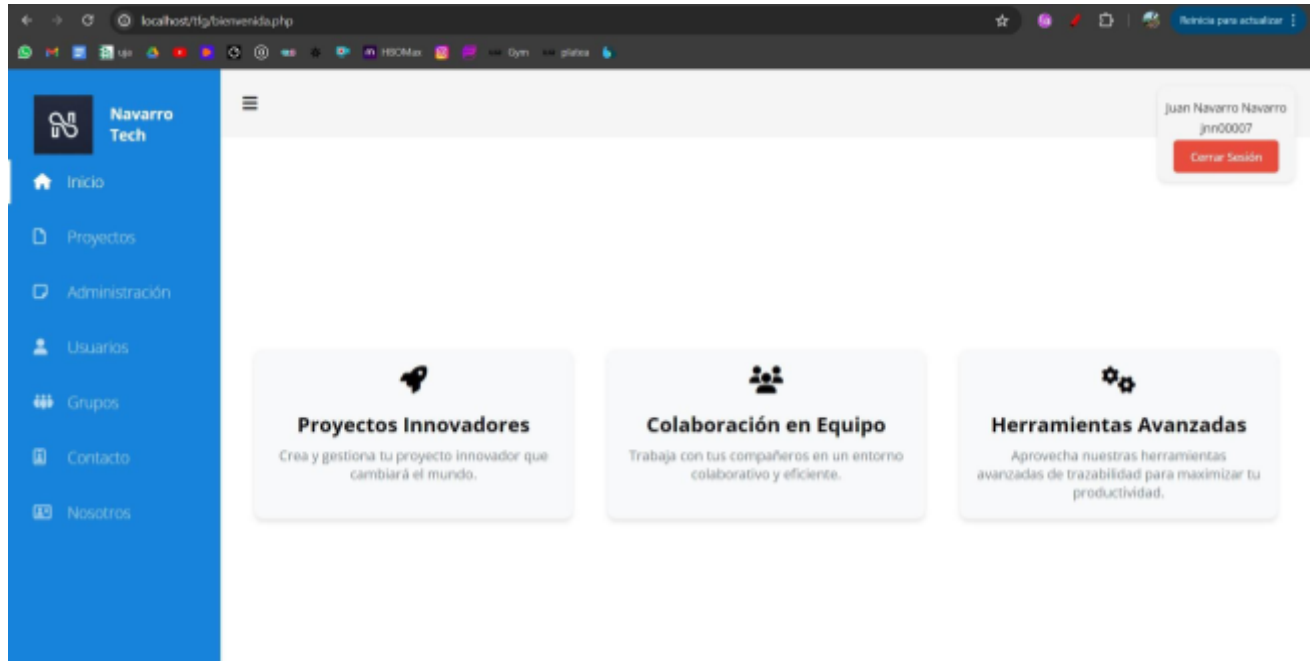


Ilustración 45: Pantalla de bienvenida

8.2.4. Menú principal y botón para cerrar sesión

El menú lateral se abre y se cierra simplemente pulsando el icono de hamburguesa de arriba.

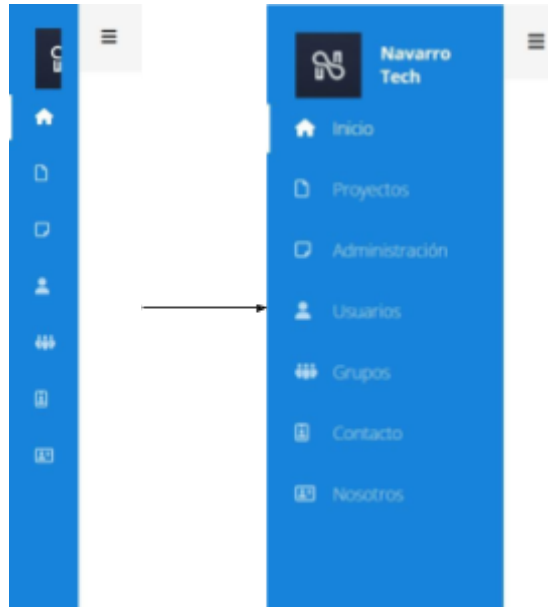


Ilustración 46: Menú principal

Al igual que el botón de cerrar sesión que al pulsar dentro del recuadro aparece.



Ilustración 47: Botón para cerrar sesión

8.2.5. Creación, modificación y eliminación de proyectos.

8.2.5.1. Creación

Pulsando sobre **“Administración”** nos desplegará 3 opciones, si pulsamos sobre **“Crear proyecto”** nos llevará a la pantalla de creación. Seguidamente meteremos los datos requeridos (si algún dato no es introducido saltará un aviso y no te dejará crearlo. Pasa lo mismo en todos los formularios de la aplicación) y pulsaremos **“Enviar”**.

localhost/tfg/crear_proyecto.php

Navarro Tech

Inicio

Proyectos

Administración

+ Crear proyecto

Modificar proyecto

Eliminar proyecto

Usuarios

Grupos

Crear Proyecto

Juan Navarro Navarro
jnn00007

Nombre del proyecto:

Escribe aquí

Descripción:

Escribe aquí

Metodología:

☐ Ágil

☐ Tradicional

Tipos de requisitos disponibles:

☐ Funcionales

☐ No Funcionales

☐ De Interfaz

☐ De Datos

☐ De Mantenimiento

Adjuntar foto portada:

Seleccionar archivo Ningún archivo seleccionado

Enviar

Ilustración 48: Pantalla creación de proyectos

8.2.5.2. Modificación

Si pulsamos sobre **“Modificar proyecto”** nos llevará a la pantalla donde aparecen todos los proyectos creados. Seguidamente elegiremos el proyecto que queremos editar y pulsaremos **“Editar”**.

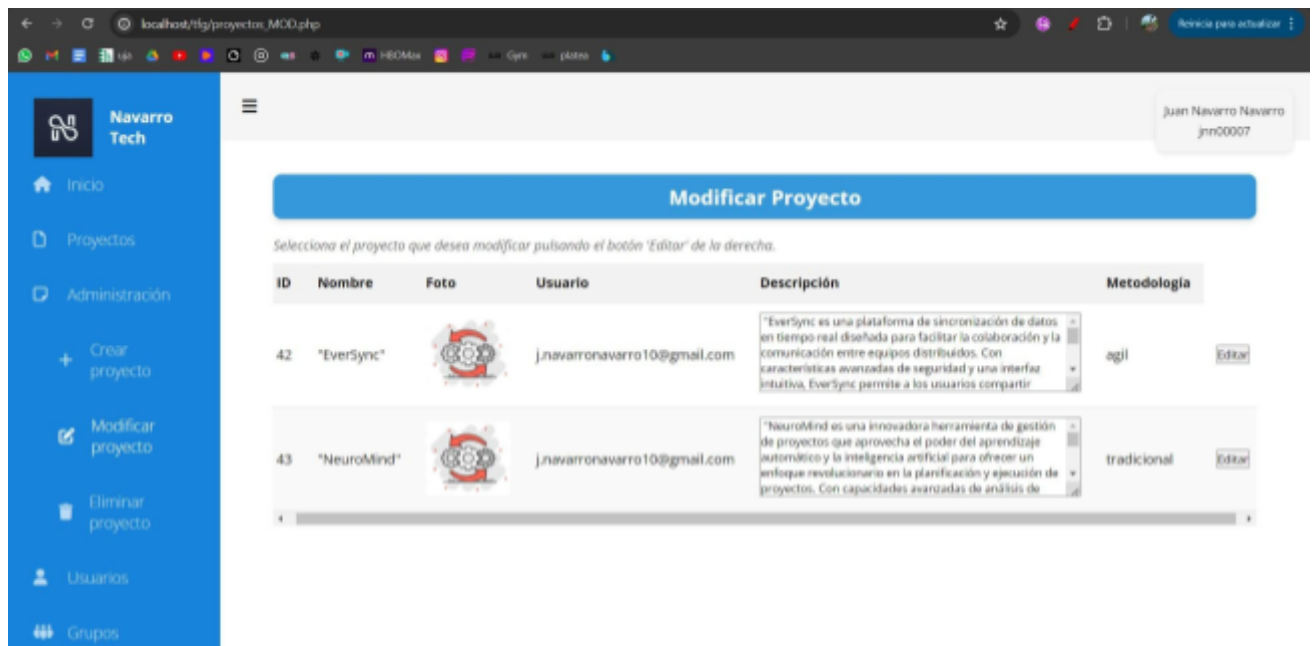


Ilustración 49: Pantalla con botón para modificación de proyectos

Este botón nos llevará a la pantalla de edición. Seguidamente meteremos los nuevos datos que queramos cambiar y pulsaremos **“Enviar”**.

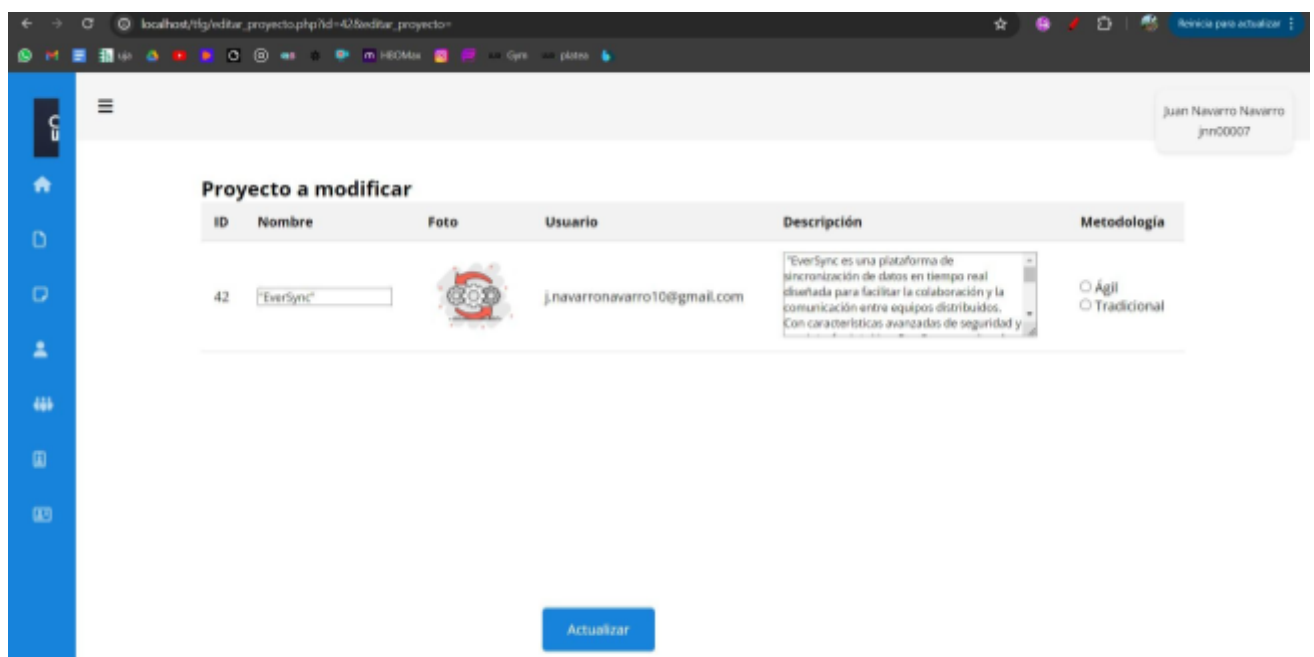


Ilustración 50: Pantalla modificación de proyectos

8.2.5.3. Eliminación

Si pulsamos sobre **“Eliminar proyecto”** nos llevará a la pantalla donde aparecen todos los proyectos creados. Seguidamente elegiremos el proyecto que queremos eliminar y pulsaremos **“Eliminar”**.

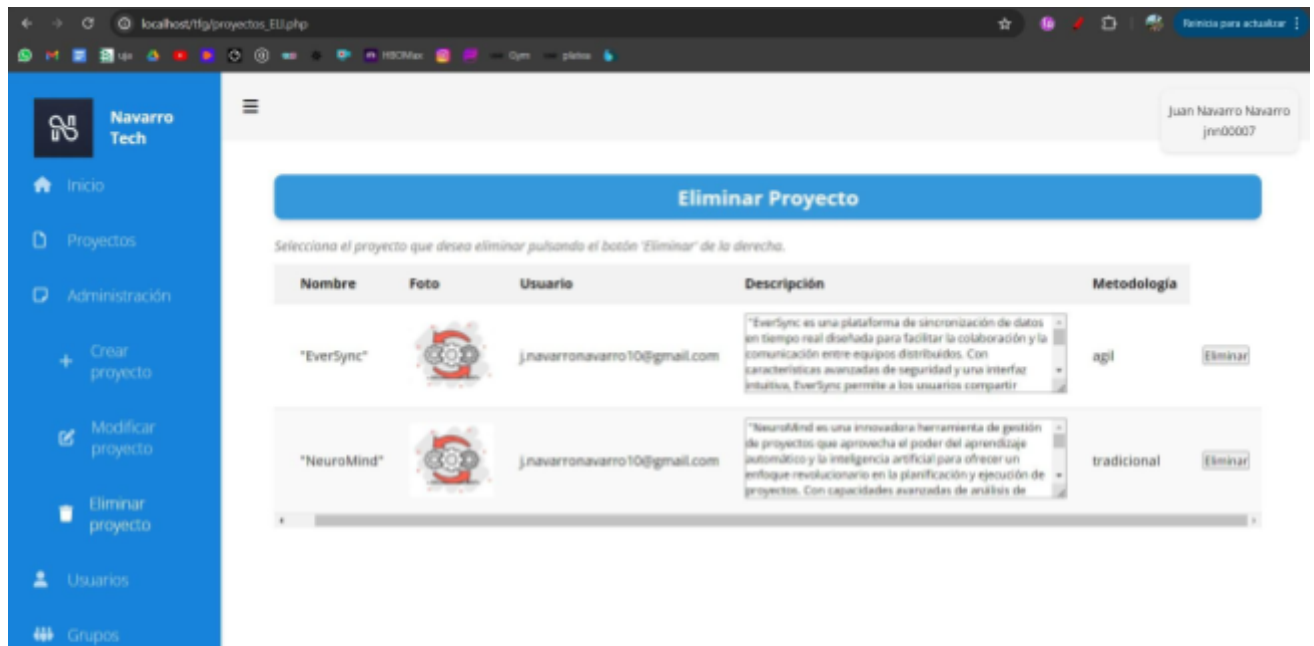
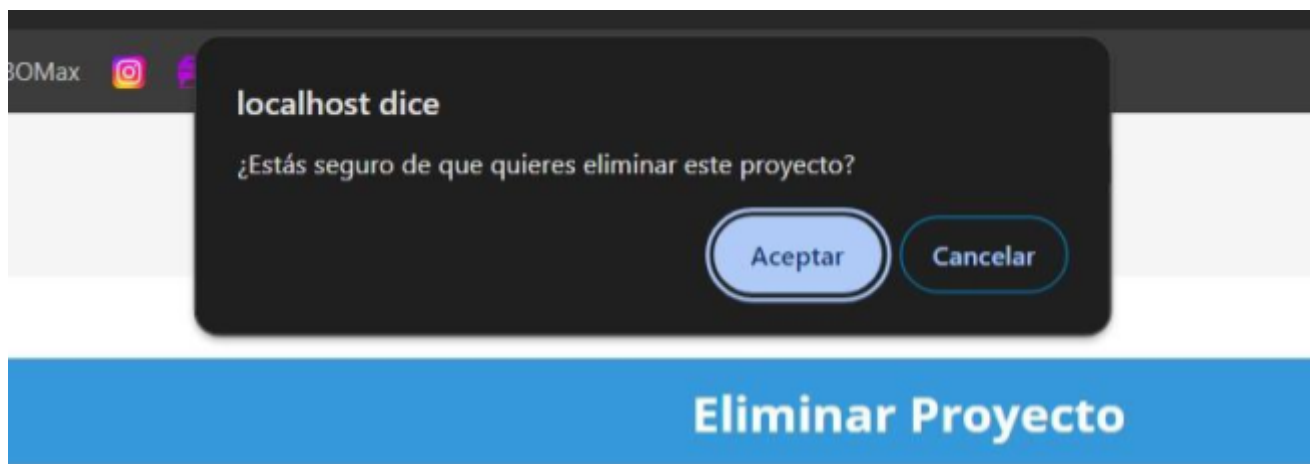


Ilustración 51: Pantalla eliminación de proyectos

Durante toda la aplicación, al pulsar algún botón que requiera un cambio en la base de datos nos saltará un mensaje de confirmación que deberemos de aceptar para que se ejecute la acción. Se hace por motivos de seguridad y asegurarse de que no se pulsa por error.



to que desea eliminar pulsando el botón 'Eliminar' de la derecha.

Ilustración 52: Alerta de cambio

8.2.6. Listado de proyectos creados

Si pulsamos sobre **“Proyectos”** nos llevará a la pantalla donde aparecen todos los proyectos creados.

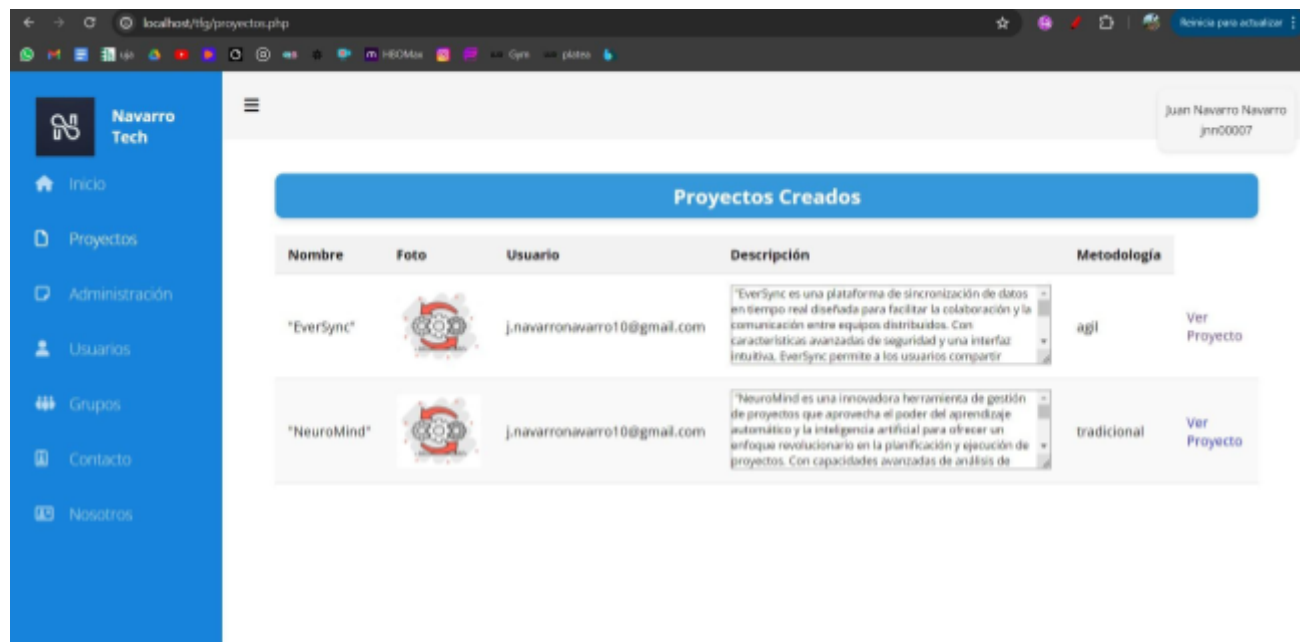


Ilustración 53: Pantalla con listado de proyectos creados

8.2.7. Creación y listado de grupos de trabajo

Pulsando sobre **“Grupos”** nos desplegará 2 opciones, si pulsamos sobre **“Crear grupo”** nos llevará a la pantalla de creación. Seguidamente meteremos los datos requeridos y pulsaremos **“Crear”**.

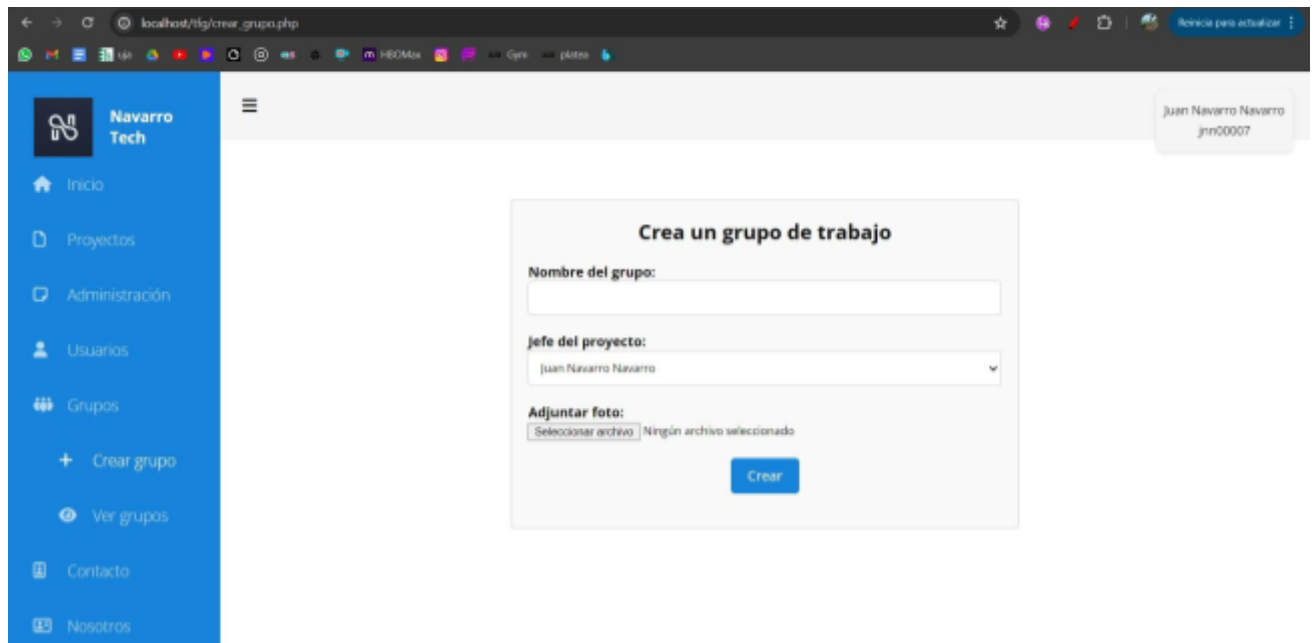


Ilustración 54: Pantalla de creación de grupos de trabajo

Si pulsamos sobre **“Ver grupos”** nos llevará a la pantalla donde aparecen todos los grupos creados.

Aquí tendremos la opción de poder asignar a cada grupo de trabajo un proyecto creado para que trabaje en él, pulsando en el desplegable, eligiendo el proyecto deseado y pulsando **“Asignar”**.

También está la opción de eliminar el grupo de trabajo que quieras, pulsando en **“Eliminar”**.

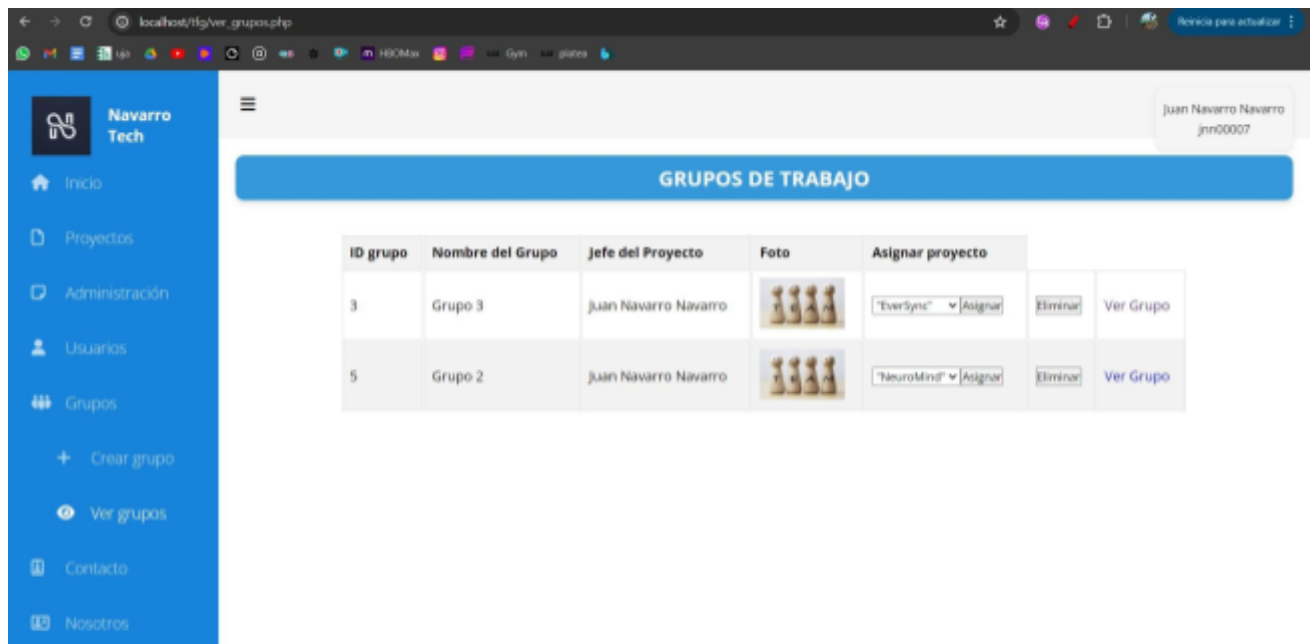


Ilustración 55: Pantalla con listado de grupos de trabajo creados

Finalmente, a la derecha encontramos la opción de **“Ver grupo”**, que nos mostrará todos los usuarios que están en ese grupo de trabajo divididos por roles.

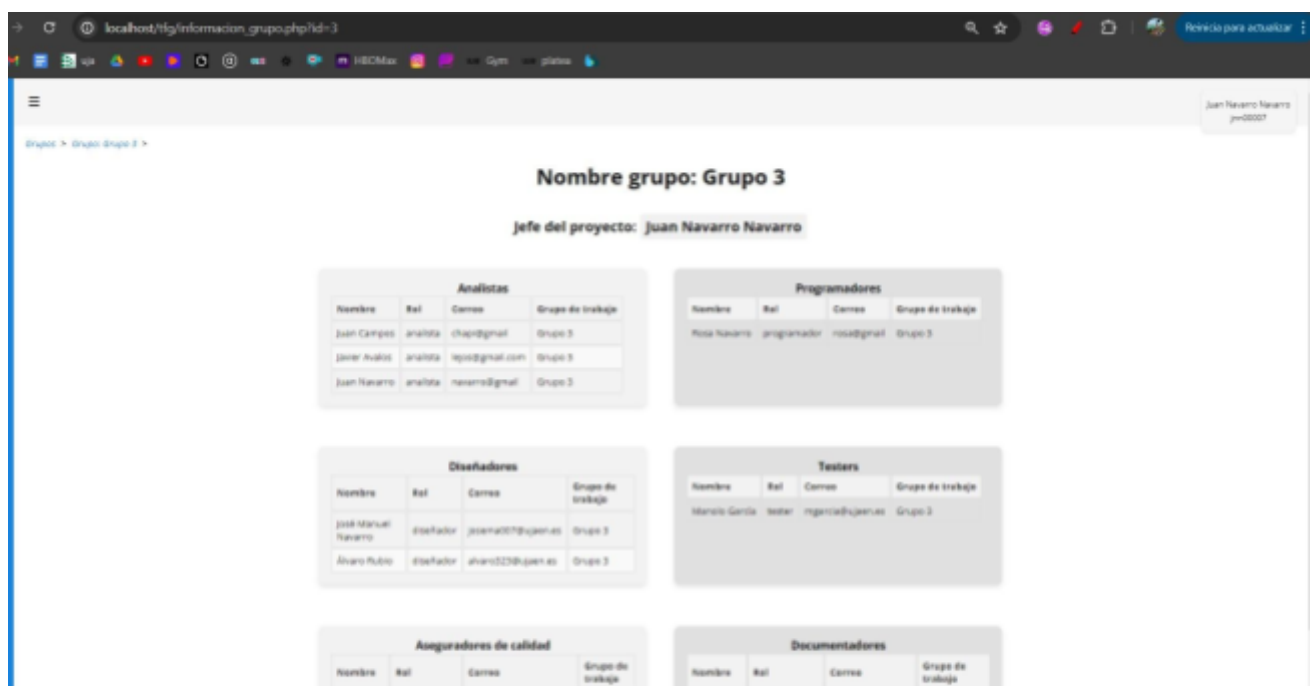


Ilustración 56: Pantalla con información del grupo de trabajo

8.2.8. Creación y listado de usuarios

Pulsando sobre **“Usuarios”** nos desplegará 2 opciones, si pulsamos sobre **“Crear usuario”** nos llevará a la pantalla de creación. Seguidamente meteremos los datos requeridos y pulsaremos **“Crear”**.

Ilustración 57: Pantalla de creación de usuario

Si pulsamos sobre **“Ver usuarios”** nos llevará a la pantalla donde aparecen todos los usuarios creados.

Aquí tendremos la opción de poder añadir cada uno a un grupo de trabajo, pulsando en el desplegable, eligiendo el grupo deseado y pulsando **“Añadir”**.

También está la opción de eliminar el usuario que quieras, pulsando en **“Eliminar”**.

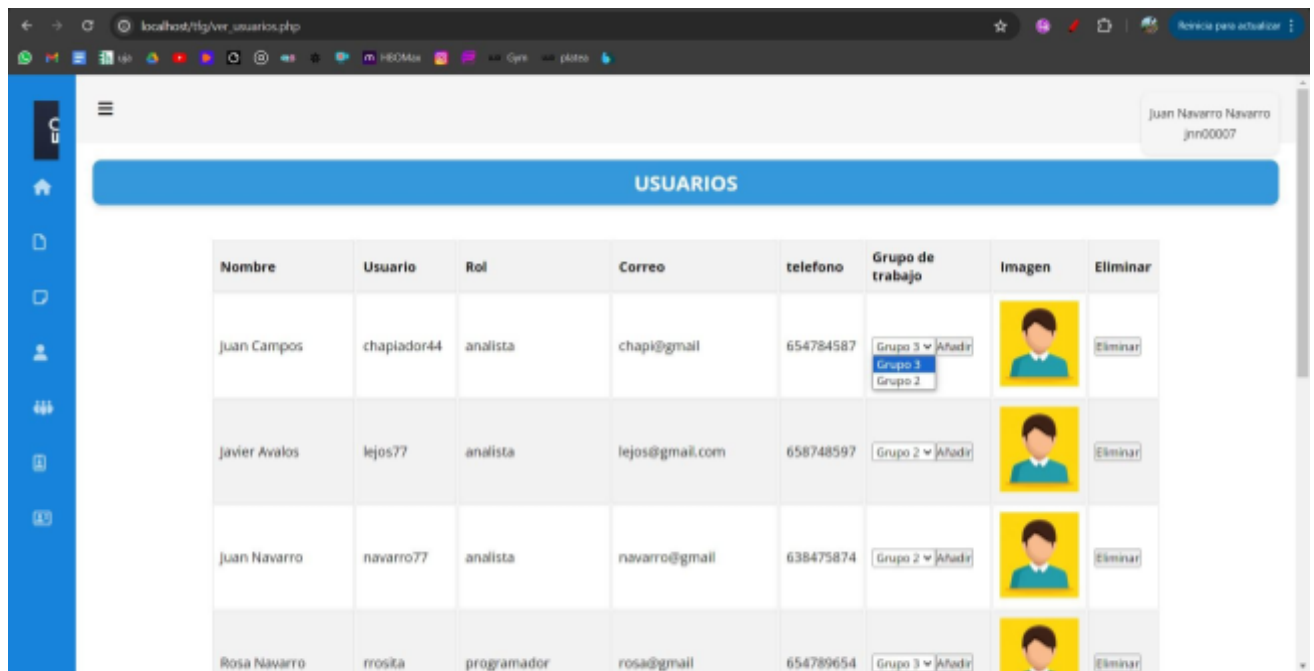


Ilustración 58: Pantalla con listado de usuarios creados

8.2.9. Ver información del proyecto

Si pulsamos sobre **“Ver proyecto”** de la pantalla de listado de proyectos nos llevará a la pantalla con la información de ese proyecto.

Aquí encontraremos por un lado los enlaces que nos llevarán al apartado de épica y de requisitos respectivamente.

Seguidamente encontramos el apartado con la información del proyecto y con la descripción de este.

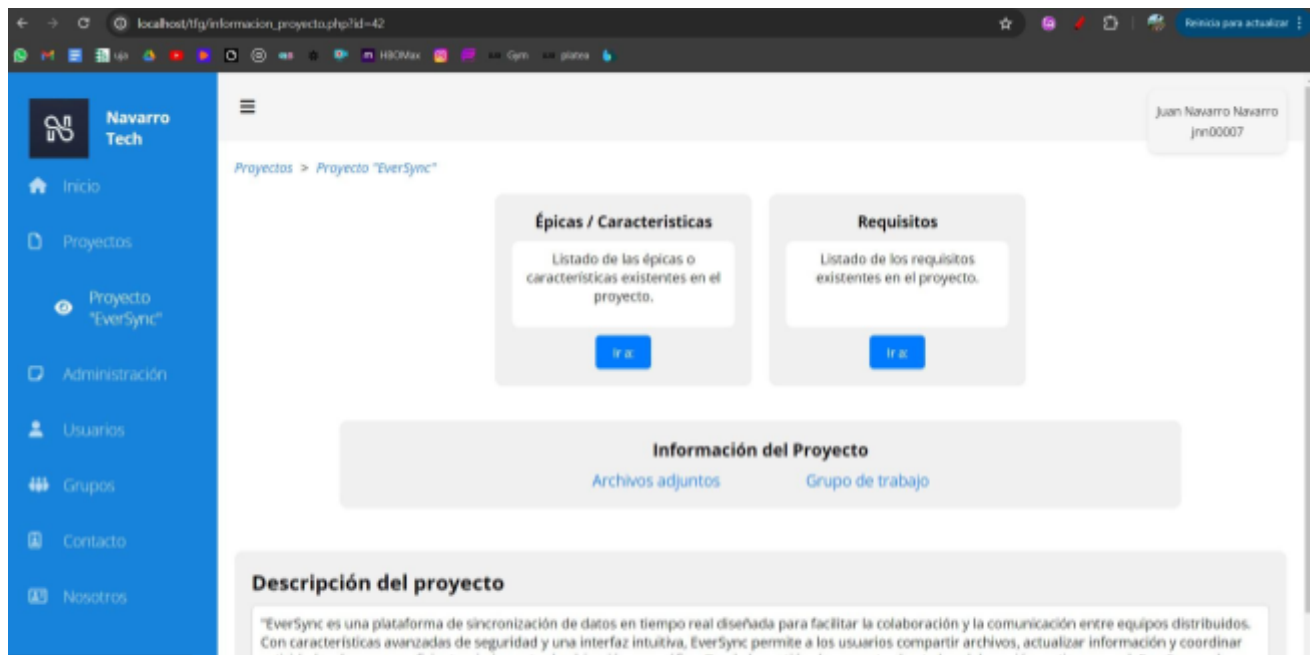


Ilustración 59: Pantalla de información de proyecto

8.2.10. Pantalla de épicas

Si pulsamos sobre **“Ir a”** de la pantalla de información de proyectos (en la parte de épicas) nos llevará a la pantalla de creación y listado de épicas de ese proyecto.

Pulsando sobre **“Añadir Épica”** nos desplegará el formulario de creación de una nueva épica. Seguidamente meteremos los datos requeridos y pulsaremos **“Crear”**.

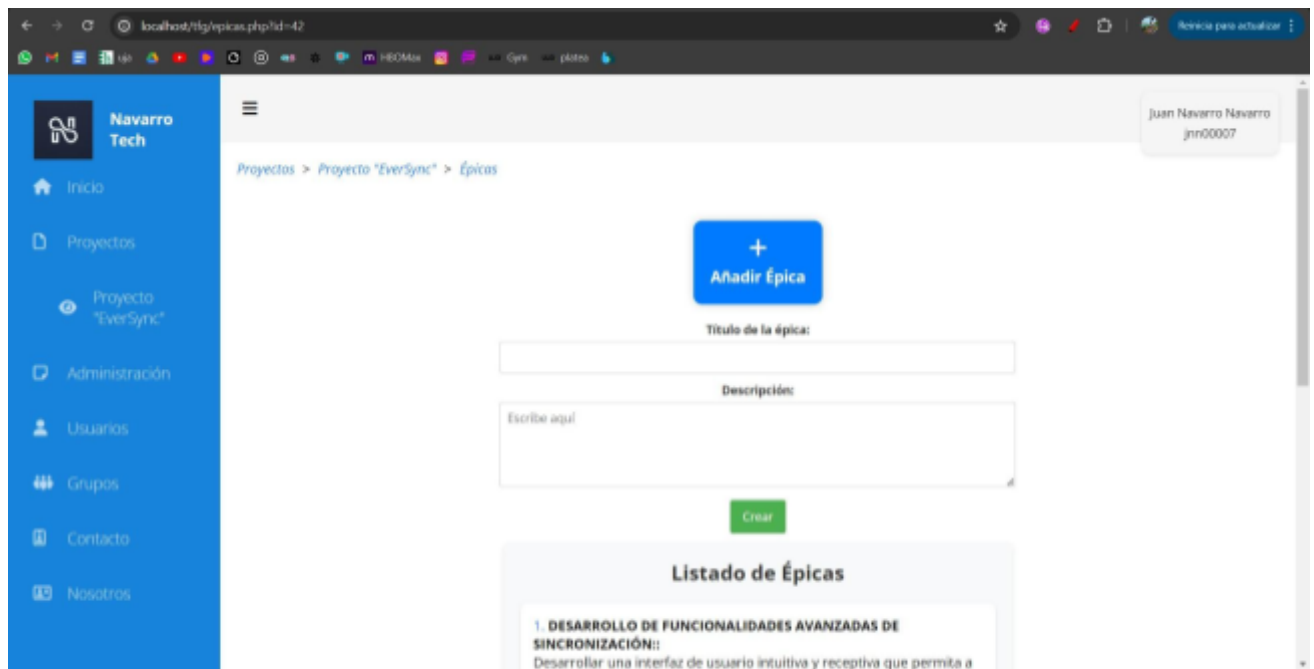


Ilustración 60: Pantalla de creación de épicas

Más abajo nos aparece el listado de épicas creadas con su descripción.

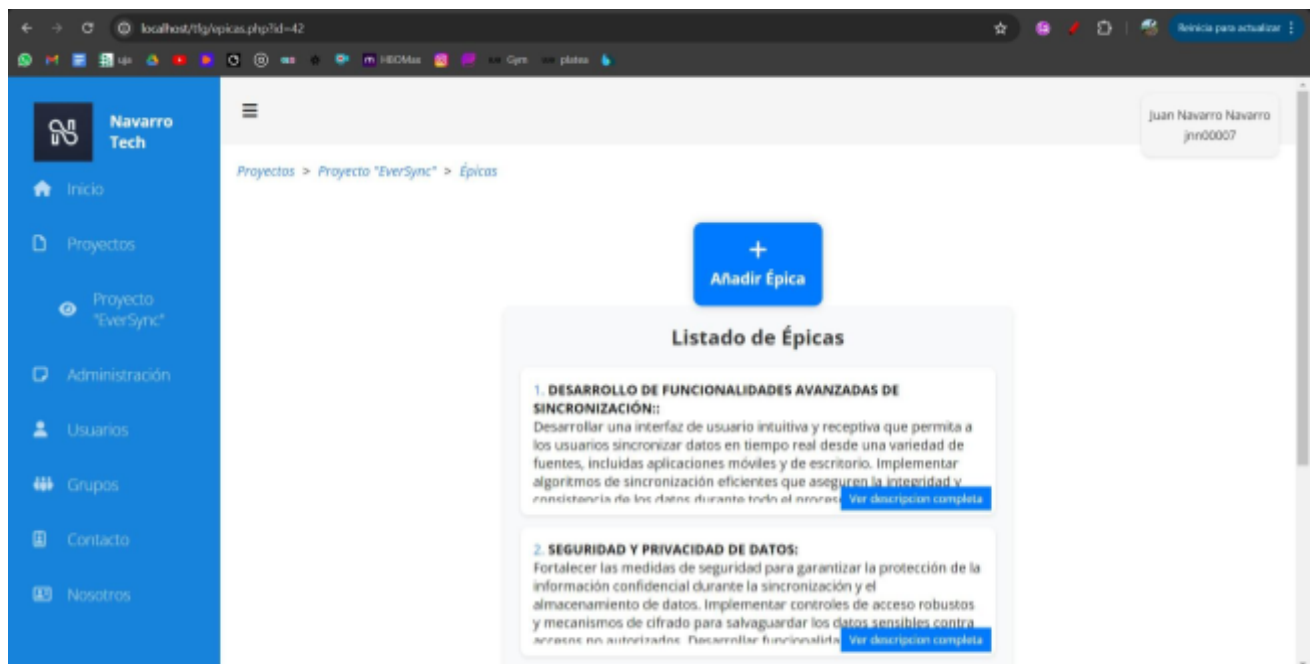


Ilustración 61: Pantalla con listado de épicas creadas

8.2.11. Pantalla de requisitos

Si pulsamos sobre **“Ir a”** de la pantalla de información de proyectos (en la parte de requisitos) nos llevará a la pantalla de creación y listado de requisitos de ese proyecto.

Pulsando sobre **“Añadir Requisito”** nos desplegará el formulario de creación de un nuevo requisito. Seguidamente meteremos los datos requeridos, seleccionaremos la épica a la que queremos que pertenezca y pulsaremos **“Crear”**.

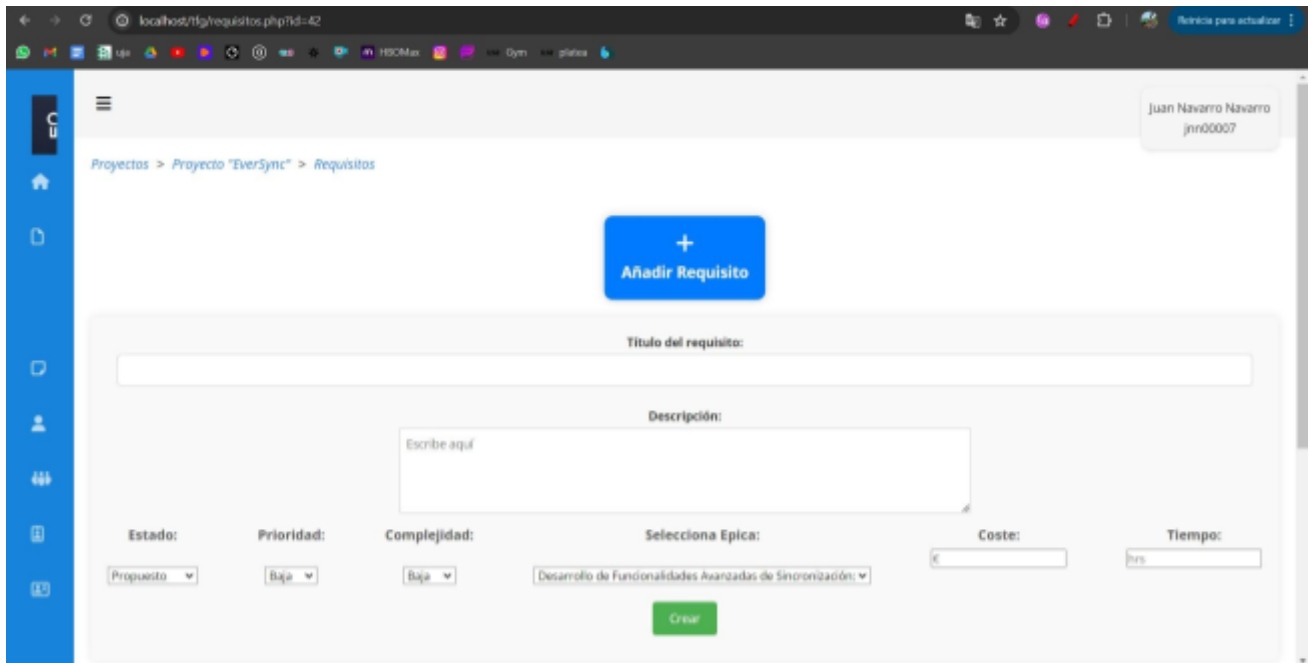


Ilustración 62: Pantalla de creación de requisitos

Más abajo nos aparece el listado de requisitos agrupados por estados y cada uno en la épica a la que pertenece.

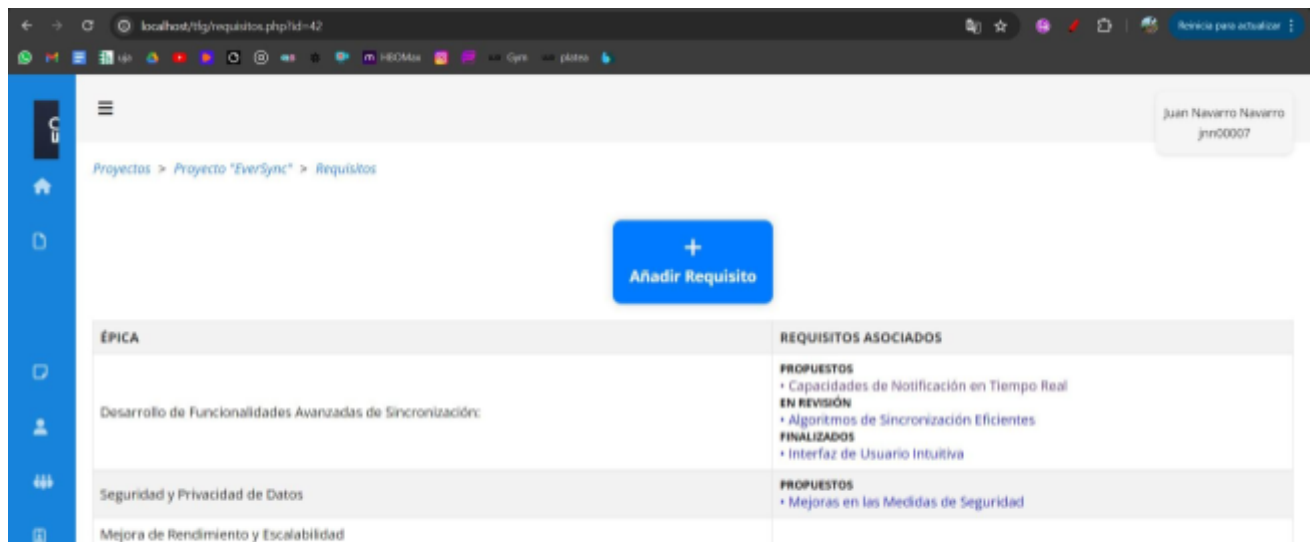


Ilustración 63: Pantalla con listado de requisitos creados

8.2.12. Trazabilidad e información del requisito

Si pulsamos sobre algún requisito de la pantalla con el listado, nos llevará a la pantalla de información del requisito donde aparece lo siguiente:

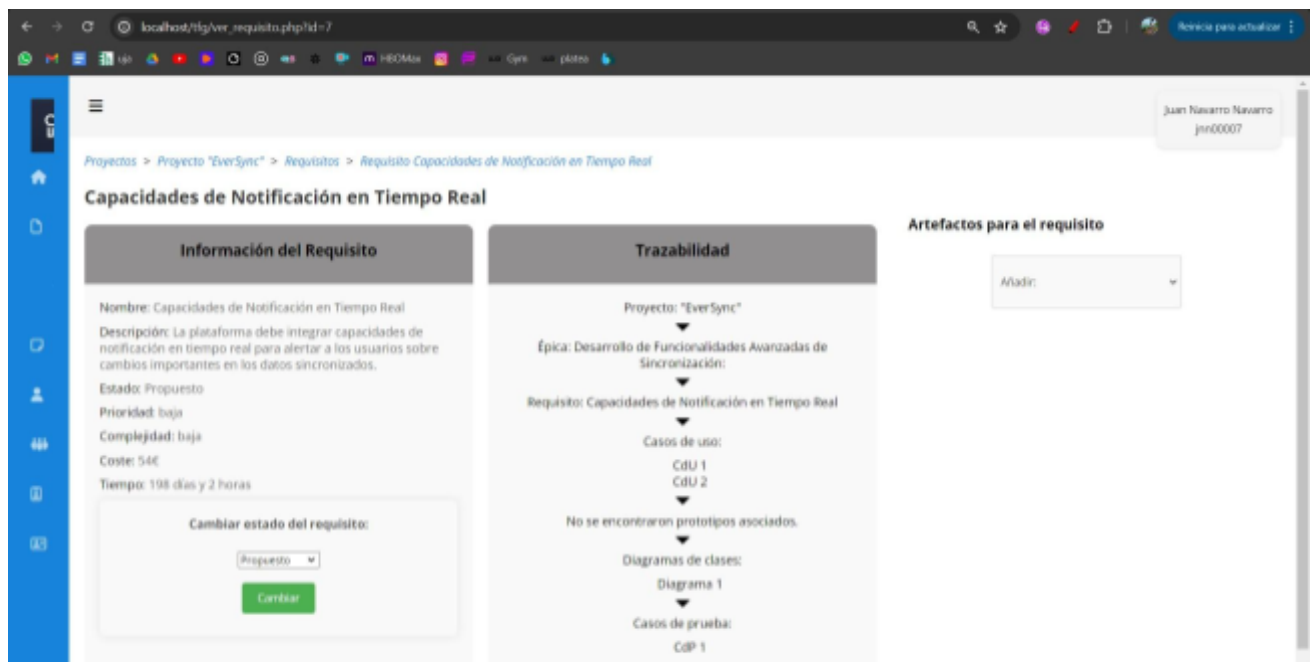


Ilustración 64: Pantalla con información del requisito

En la parte izquierda encontraremos toda la información de ese requisito junto con una casilla para cambiar el estado de este, en función de por las fases en las que vaya pasando.

The diagram illustrates the process of changing the status of a requirement. It consists of two main parts connected by a downward arrow.

The top part shows a form titled "Cambiar estado del requisito:". Below the title is a dropdown menu with the following options: "Propuesto", "Propuesto" (highlighted), "En revisión", "En desarrollo", and "FINALIZADO".

The bottom part shows the same form after the status has been changed. The dropdown menu now displays "FINALIZADO". Below the dropdown menu is a green button labeled "Cambiar".

Ilustración 65: Cuadro de cambio de estado del requisito

En la parte derecha nos encontramos con la parte de creación de artefactos. Si pulsamos sobre el desplegable **“Añadir”** nos desplegará las 4 opciones de artefactos que podemos crear y añadir para ese requisito.

Al pulsar sobre alguno de ellos se mostrará el formulario para su creación, y finalmente al pulsar **“crear”** se añadirá y actualizará el apartado de trazabilidad de ese requisito.

The diagram illustrates the process of creating an artifact. On the left, a box titled "Artefactos para el requisito" contains a dropdown menu labeled "Añadir:". The dropdown is open, showing a list of options: "Añadir:", "Caso de uso", "Prototipo de interfaz de usuario", "Diagrama de clases", and "Caso de prueba". An arrow points from this dropdown to a larger form on the right titled "Nuevo Caso de Uso". This form has a dropdown at the top labeled "Caso de uso". Below it are four text input fields: "Título:", "Descripción:", "Actores:", and "Flujo básico:". The "Flujo básico:" field has a placeholder text "Escribe el flujo básico paso a paso...". At the bottom right of the form is a green button labeled "Crear".

Ilustración 66: Formulario de creación de artefactos

Finalmente, en la parte central, tenemos la **trazabilidad** del requisito. En ella observamos:

- El proyecto al que pertenece.
- La épica a la que pertenece.
- El requisito en el que estamos.
- Los casos de uso asociados.
- Los prototipos de interfaz de usuario asociados.
- Los diagramas de clases asociados.
- Los casos de prueba asociados.

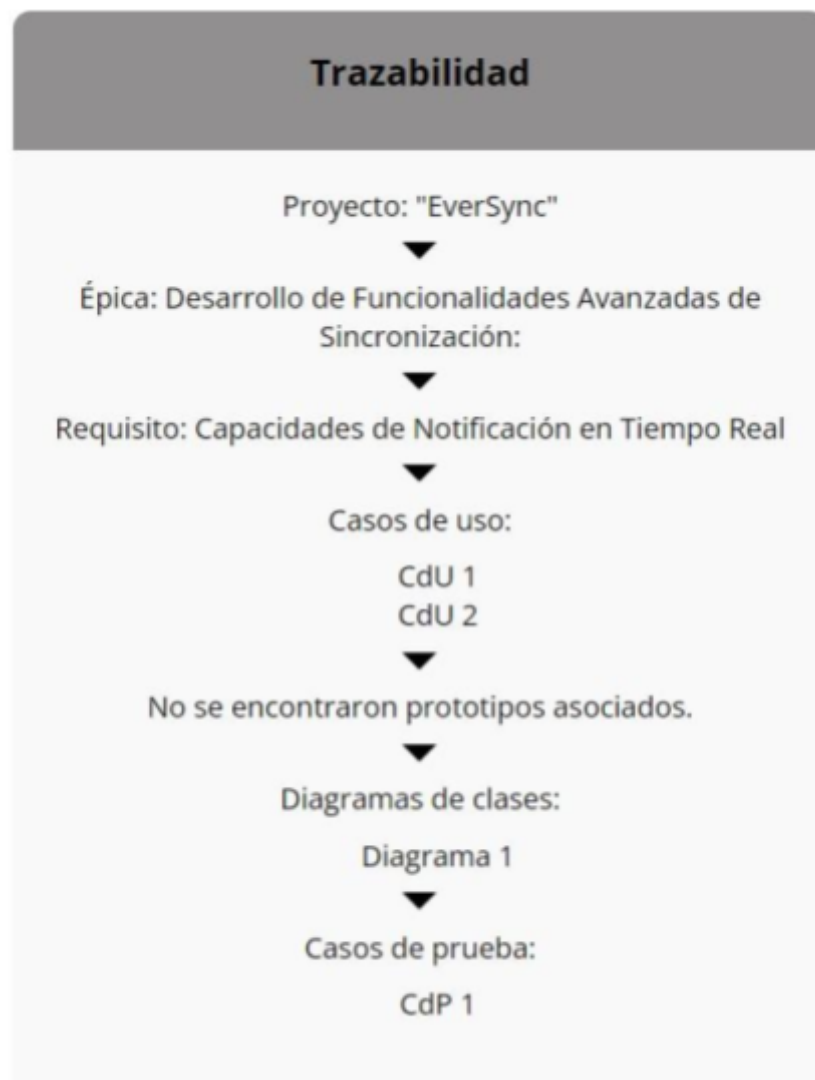


Ilustración 67: Cuadro con la trazabilidad del requisito

9. CONCLUSIÓN FINAL

La idea principal detrás de esta aplicación es hacer tu vida más fácil al gestionar proyectos, requisitos, usuarios y grupos de trabajo, mediante el uso de la trazabilidad.

Desarrollar esta herramienta ha sido un viaje lleno de aprendizaje y desafíos, pero estoy muy contento con el resultado final. Ahora tienes una aplicación que no solo te ayuda a organizar mejor tu trabajo, sino que también mejora la colaboración dentro de tu equipo.

Al usar esta herramienta, podrás ver cómo todo encaja de manera ordenada, desde los requisitos iniciales hasta la implementación final. Esto no solo ahorra tiempo, sino que también reduce el riesgo de que algo se pase por alto. La trazabilidad de requisitos es una parte crucial de cualquier proyecto de software exitoso, y esta aplicación está aquí para asegurarse de que todo esté bajo control. Me ha permitido comprender el porqué de que sea tan utilizado, y es que, en mi opinión, facilita mucho el desarrollo de plataformas empresariales debido a que es simple y elimina mucho código.

Recuerda que la clave del éxito en cualquier proyecto es la buena comunicación y organización, y eso es exactamente lo que esta herramienta pretende facilitar.

Trabajar con tecnologías como PHP, CSS, JavaScript y MySQL, así como utilizar herramientas como Visual Studio Code y XAMPP, ha sido una oportunidad para mejorar mis habilidades técnicas. He aprendido a integrar diferentes tecnologías de manera efectiva y a resolver problemas complejos que surgían durante el desarrollo.

Desarrollar esta herramienta de trazabilidad de requisitos ha sido una experiencia increíblemente enriquecedora para mí, tanto a nivel personal como profesional. No solo me ha permitido aplicar y profundizar mis conocimientos en ingeniería de software, sino que también me ha hecho crecer como ingeniero.

¡Buena suerte con tus proyectos y feliz trazabilidad de requisitos!

10. BIBLIOGRAFÍA Y REFERENCIAS

1. Nuvaweb: Página con información para la trazabilidad de requisitos software, <https://nuvaweb.com/ingenieria-de-requisitos/#gs.bfg20j>, (fecha de consulta: 30-11-2023).
2. Slideshares: Artículo con información para la trazabilidad de requisitos software, <https://es.slideshare.net/slideshow/ingeniera-de-requisitos-y-la-ingeniera-d-e-requerimientos/71394962>, (fecha de consulta: 30-11-2023).
3. revinformatica: Artículo con información para la trazabilidad de requisitos software, <https://revinformatica.sld.cu/index.php/rcim/article/view/375/532>, (fecha de consulta: 30-11-2023).
4. Sedici: Artículo con información para la trazabilidad de requisitos software, https://sedici.unlp.edu.ar/bitstream/handle/10915/4057/2_-Ingenier%C3%ADa_de_requerimientos.pdf?sequence=4&isAllowed=y, (fecha de consulta: 30-11-2023).
5. Artículo con información para la trazabilidad de requisitos software, <https://gc.scalahed.com/recursos/files/r161r/w24792w/IRADS/INGRequisitos.pdf>, (fecha de consulta: 30-11-2023).
6. Digital guide: Página de ayuda para la instalación y primeros pasos de Xampp, <https://www.ionos.es/digitalguide/servidores/herramientas/instala-tu-servidor-local-xampp-en-unos-pocos-pasos/>, (fecha de consulta: 14-11-2023).
7. ReqView: Requirements Management Tool, <https://www.reqview.com/>, (fecha de consulta: 02-12-2023).
8. SpiraPlan: Requirements Management Tool, <https://www.inflectra.com/Products/SpiraPlan/>, (fecha de consulta: 02-12-2023).
9. Accompa: Requirements Management Tool, <https://web.accompa.com/>, (fecha de consulta: 02-12-2023).
10. Codebeamer: Requirements Management Tool, <https://www.ptc.com/en/products/codebeamer>, (fecha de consulta: 02-12-2023).
11. Proyectosagiles: Página con información para el desarrollo iterativo e incremental, <https://proyectosagiles.org/desarrollo-iterativo-incremental/>, (fecha de consulta: 12-01-2024).
12. Infor.uva: Documento con información para la estimación mediante puntos de función, <https://www.infor.uva.es/~jvalvarez/docencia/ptema3is1.pdf>, (fecha de consulta: 08-06-2024).
13. Artículo con información para la estimación mediante puntos de función, <https://manuel.cillero.es/doc/metodologia/metrica-3/tecnicas/tecnicas-de-estimacion/metodo-mark-ii/>, (fecha de consulta: 08-06-2024).
14. LocalHost: Host local para la base de datos usada en la aplicación, <http://localhost/phpmyadmin/index.php?route=/database/>, (fecha de consulta: durante todo el proyecto).
15. Blog Visual-paradigm: Página con información para el uso de visual paradigm,

- <https://blog.visual-paradigm.com/es/include-and-extend-use-cases/>,
(fecha de consulta: durante todo el proyecto).
16. Smartmockups: Página con información para la creación de mockups,
<https://smartmockups.com/es>, (fecha de consulta: durante todo el
proyecto).
17. Plantuml: Página con información para la creación de uml,
<https://plantuml.com/es/>, (fecha de consulta: durante todo el proyecto)
18. Fontawesome: Página con los iconos usados durante la programación de
la herramienta, <https://fontawesome.com/>, (fecha de consulta: durante
todo el proyecto).