Universidad de Valladolid

Blockchain y Project Management. Diseño e Implementación de Contratos Inteligentes en PM².

Daniel Fernández López

MÁSTER EN DIRECCIÓN DE PROYECTOS
Departamento De Organización De Empresas Y C.I.M.
Universidad De Valladolid
España

máster DP UVa

INSISOC
SOCIAL SYSTEMS ENGINEERING CENTRE
2025

Blockchain y Project Management. Diseño e Implementación de Contratos Inteligentes en PM².

Daniel Fernández López

MÁSTER EN DIRECCIÓN DE PROYECTOS
Departamento De Organización De Empresas Y C.I.M.
Universidad De Valladolid

Valladolid, Julio 2025

Tutores
Joaquín Nicolás Adiego Rodríguez
Fernando Acebes Senovilla
Teresa Natalia Martín Cruz

AGRADECIMIENTOS

A mi familia y amigos, por estar siempre ahí.
A quienes han compartido esta etapa conmigo, dentro y fuera del aula.
Gracias a mis tutores por su apoyo y orientación.

RESUMEN

La metodología PM² ofrece un marco estructurado y adaptable para la gestión de proyectos, pero su aplicación práctica sigue anclada a herramientas tradicionales que limitan la trazabilidad y automatización. Este trabajo propone una solución basada en contratos inteligentes sobre *blockchain* para digitalizar artefactos clave de PM² y reforzar la transparencia, seguridad y eficiencia del proceso.

Se han identificado los artefactos con mayor viabilidad *on-chain* y se ha desarrollado un sistema funcional que integra un contrato inteligente en Ethereum y una interfaz web en React. El sistema permite gestionar solicitudes de cambio de forma automatizada, verificable y auditable.

La solución ha sido validada en entorno de pruebas y evaluada desde una perspectiva técnica, económica y ambiental. Los resultados demuestran el potencial de *blockchain* para transformar la gestión de proyectos dentro del marco PM², ofreciendo una alternativa moderna y descentralizada alineada con los principios de gobernanza, control y mejora continua.

Palabras clave

Gestión de Proyectos, *Blockchain*, Contratos Inteligentes, Digitalización de Procesos, Metodología PM^2.

ABSTRACT

The PM² methodology offers a structured and adaptable framework for project management, yet its practical application remains tied to traditional tools that limit traceability and automation. This work proposes a solution based on smart contracts deployed on blockchain to digitize key PM² artefacts and enhance transparency, security, and process efficiency.

The most suitable artefacts for on-chain implementation were identified, and a functional system was developed, integrating a smart contract on Ethereum and a web interface built with React. The system enables the automated, verifiable, and auditable management of change requests.

The solution was validated in a test environment and evaluated from technical, economic, and environmental perspectives. Results demonstrate the potential of blockchain to transform project management within the PM² framework, offering a modern and decentralized alternative aligned with core principles of governance, control, and continuous improvement.

Keywords

Project Management, Blockchain, Smart Contracts, Process Digitalization, PM² Methodology.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

La dirección de proyectos se enfrenta actualmente a un entorno caracterizado por una creciente complejidad, una mayor demanda de transparencia y una acelerada transformación digital. La evolución de las metodologías de gestión ha respondido en gran medida a estos desafíos, desarrollando marcos como PMBOK, PRINCE2, Lean o Agile, y más recientemente, adaptaciones como PM², que buscan combinar solidez estructural con flexibilidad operativa. Sin embargo, en paralelo al desarrollo metodológico, han emergido tecnologías con un potencial disruptivo aún poco explorado en este ámbito, como es el caso de la *blockchain*.

Aunque la transformación digital lleva años en marcha y ciertas metodologías ya han comenzado a adoptar herramientas digitales (por ejemplo, flujos de trabajo automatizados, repositorios compartidos o firmas electrónicas), muchas de estas soluciones siguen operando bajo arquitecturas centralizadas que presentan limitaciones en términos de trazabilidad, interoperabilidad y seguridad. Aquí es donde *blockchain* introduce un nuevo paradigma: una infraestructura distribuida capaz de registrar información de forma inmutable, transparente y verificable, sin depender de intermediarios de confianza.

Esta tecnología, ya consolidada en sectores como las finanzas o la logística, empieza a despertar interés en el ámbito de la gestión de proyectos, no tanto por reemplazar los marcos metodológicos existentes, sino por complementarlos mediante la automatización de procesos clave y la mejora de la gobernanza documental. El problema, sin embargo, radica en la escasez de propuestas prácticas que demuestren cómo integrar de forma efectiva esta tecnología dentro de una metodología estructurada, respetando su lógica interna y sin añadir complejidad innecesaria.

En este contexto, este trabajo propone una línea concreta de investigación: la implementación parcial de *blockchain* en artefactos clave de la metodología PM², centrando el desarrollo en el artefacto *Change Log*, y utilizando contratos inteligentes escritos en Solidity y desplegados en un entorno de pruebas. La elección de PM² responde a su carácter público, su claridad documental y su orientación a proyectos europeos y colaborativos, lo que la convierte en un entorno ideal para validar la aplicabilidad de *blockchain* en la gestión de proyectos.

A lo largo del proyecto se aborda una revisión crítica de las metodologías más relevantes, un análisis detallado del marco PM² y una selección razonada de artefactos con potencial de ser automatizados. A partir de ahí, se diseña e implementa una solución técnica que permite registrar cambios en el proyecto de forma inmutable, verificable y transparente, con el objetivo de mejorar la trazabilidad y la fiabilidad del control de cambios. Aunque la implementación se limita a un prototipo funcional, los resultados obtenidos permiten vislumbrar un modelo escalable y replicable para otros artefactos o contextos similares.

Este trabajo, por tanto, no sólo se enmarca en la convergencia entre tecnología emergente y gestión profesionalizada, sino que también aporta una propuesta concreta y viable para avanzar en la digitalización de procesos clave dentro de la dirección de proyectos, ofreciendo una base sobre la que construir soluciones más ambiciosas en el futuro.

Objetivo del Proyecto

El presente Trabajo de Fin de Máster tiene como objetivo principal diseñar un modelo de integración de la tecnología *blockchain*, mediante contratos inteligentes programados en Solidity y desplegados en la red Ethereum, dentro del marco metodológico de gestión de proyectos PM². El propósito es

demostrar cómo esta integración puede mejorar la eficiencia operativa, la transparencia de los procesos, la trazabilidad de decisiones y la seguridad documental en el ciclo de vida del proyecto. A través de esta propuesta, se busca aportar una solución práctica y viable que aproveche las capacidades de *blockchain* para optimizar la gestión de artefactos clave en proyectos estructurados bajo PM².

Alcance del Proyecto

Este Trabajo Fin de Máster se enmarca en la intersección entre la gestión de proyectos y la tecnología *blockchain*, centrando su alcance en la exploración del potencial de esta tecnología para transformar ciertos procesos y artefactos dentro de metodologías formales de dirección de proyectos. En particular, se aborda un análisis comparativo de diferentes enfoques metodológicos con el objetivo de identificar cuál de ellos resulta más adecuado para una integración efectiva con contratos inteligentes. La investigación se enfoca en la metodología PM² por su claridad estructural y por ofrecer un marco documental propicio para la digitalización y automatización.

Dentro de este marco, el alcance técnico se limita a la identificación de artefactos clave y a la implementación práctica de uno de ellos (*Change Log*) mediante un contrato inteligente desarrollado en *Solidity* y desplegado en un entorno local. No se aborda, sin embargo, la integración real en entornos empresariales, ni el desarrollo completo de una suite de herramientas *blockchain* para todos los artefactos de PM². Asimismo, no se contempla un análisis legal o normativo profundo sobre la aplicación de *blockchain* en la gestión documental, ni se abordan aspectos económicos o de viabilidad financiera para una adopción a escala organizacional.

El proyecto, por tanto, se centra en la validación conceptual y técnica a nivel de prototipo, dejando abiertas múltiples líneas de trabajo para investigaciones futuras que contemplen pruebas piloto, análisis coste-beneficio, integración con plataformas de gestión de proyectos existentes o estudio de casos reales.

Motivación del Proyecto

La motivación de este proyecto surge de la convergencia entre dos realidades: por un lado, el creciente interés institucional y académico en la tecnología *blockchain* como herramienta disruptiva para mejorar procesos organizativos; y por otro, la falta de estudios que exploren de forma concreta su aplicación dentro de metodologías formales de gestión de proyectos como PM². A pesar del potencial demostrado de *blockchain* en áreas como trazabilidad, seguridad y automatización, su integración específica en marcos metodológicos como PM² continúa siendo un terreno poco explorado. Esta oportunidad de investigación, sumada al interés por aportar soluciones reales a la transformación digital de la gestión de proyectos, justifica y sustenta el desarrollo de este trabajo.

En este contexto, este TFM se encuadra dentro de una de las líneas de investigación de INSISOC, Grupo de Investigación Reconocido (GIR) de la Universidad de Valladolid. En concreto, dentro de la línea que tiene como objetivo la introducción de la *blockchain* en aplicaciones relacionadas con la cadena de suministro o la dirección de proyectos. Esta investigación está alineada con trabajos previos realizados como, por ejemplo, Sobrino et al. [1], Sobrino, M., Pajares, J., et al. [2], Adiego, J. et al. [3], o el proyecto europeo citado en [4].

Competencias Aplicadas

El presente Trabajo de Fin de Máster se enmarca plenamente dentro de las competencias y objetivos formativos establecidos por el Máster en Dirección de Proyectos de la Escuela de Ingenierías Industriales de la Universidad de Valladolid, permitiendo su aplicación real y práctica en un contexto innovador y tecnológico.

En primer lugar, este trabajo ha supuesto una oportunidad para desarrollar competencias profesionales específicas en la dirección y gestión de proyectos, aplicando metodologías formales como PM² y explorando su adaptación a tecnologías emergentes como *blockchain*. Esta integración exige no solo un conocimiento técnico profundo de los marcos de gestión, sino también la capacidad para analizar, seleccionar y aplicar herramientas y metodologías apropiadas según el tipo de proyecto y sus requisitos específicos, lo que refleja una aplicación directa de la formación general y específica que persigue el máster.

Asimismo, el diseño y desarrollo de contratos inteligentes para artefactos clave de PM² ha requerido capacidades analíticas, organizativas y de resolución de problemas multidisciplinares, alineadas con el perfil profesional que promueve la EII: ingenieros con visión sistémica, orientación a la innovación y capacidad de adaptación tecnológica.

A lo largo del desarrollo del trabajo también se han puesto en práctica habilidades transversales clave como la gestión autónoma del tiempo, el pensamiento crítico, la toma de decisiones y la capacidad de comunicación escrita y técnica, indispensables para liderar o integrarse en equipos de trabajo en entornos complejos y digitalizados.

El enfoque del TFM (que combina metodología estructurada, innovación tecnológica y análisis aplicado) responde igualmente a la necesidad de familiarizarse con entornos empresariales organizados por proyectos, permitiendo al estudiante comprender y modelar sus dinámicas, sus riesgos y sus necesidades tecnológicas reales.

Estructura del Documento

El presente trabajo se organiza en cinco capítulos, cada uno de los cuales aborda una dimensión específica del proyecto, desde la revisión conceptual hasta la implementación técnica y el análisis de resultados:

Capítulo 1 – Revisión de literatura: ofrece un recorrido por los principales conceptos relacionados con la dirección de proyectos, la evolución hacia la digitalización y las tecnologías emergentes asociadas, como Web3, *blockchain* y contratos inteligentes. También incluye una comparativa de metodologías de gestión de proyectos y su potencial integración con *blockchain*, con especial atención a PM².

Capítulo 2 – Metodología: se analiza en profundidad la metodología PM², sus principios, procesos y artefactos. A continuación, se justifica la elección de la red Ethereum y el lenguaje Solidity, y se presenta un análisis detallado de los artefactos PM² según su idoneidad para ser implementados mediante contratos inteligentes.

Capítulo 3 – Implementación: se describe cómo se ha llevado a cabo la digitalización mediante *blockchain* de los artefactos más relevantes: *Project Charter*, *Change Log*, *Change Request Form*,

Deliverable Acceptance Form y *Deliverable Acceptance Plan*. Para cada uno, se expone la problemática que resuelve, su lógica *on-chain* y los beneficios obtenidos.

Capítulo 4 – Diseño e implementación técnica del sistema basado en contratos inteligentes: detalla el diseño del contrato inteligente, su estructura de datos, funciones clave, validaciones y control de acceso. También se presenta la aplicación *frontend* desarrollada para interactuar con el sistema, así como las pruebas funcionales realizadas en entorno de *test*.

Capítulo 5 – Discusión y conclusiones: se analizan los resultados obtenidos, sus limitaciones y riesgos, junto con un estudio económico, una evaluación del impacto medioambiental y propuestas de mejora futura. El trabajo concluye con una reflexión final sobre el valor de esta propuesta en el contexto de la gestión de proyectos moderna.

Capítulo 1 - Revisión de Literatura

La gestión de proyectos se encuentra en un proceso de transformación impulsado por la digitalización y la adopción de tecnologías emergentes. En este contexto, resulta clave revisar cómo herramientas como Web3 y *blockchain* están comenzando a integrarse en las metodologías de dirección de proyectos.

Este capítulo presenta una revisión de la evolución de la gestión de proyectos hacia modelos más automatizados y transparentes, analiza los fundamentos del *blockchain* y los contratos inteligentes, y examina su aplicabilidad en metodologías como *Waterfall*, PMBOK, ICB, PRINCE2, Lean, Agile y PM². Esta base teórica servirá como punto de partida para el desarrollo del modelo propuesto en este trabajo.

1.1 Dirección de Proyectos y su Evolución hacia la Digitalización

La dirección de proyectos puede definirse como la aplicación de conocimientos, habilidades, herramientas y técnicas para planificar, ejecutar, controlar y cerrar proyectos, logrando cumplir objetivos específicos dentro de restricciones determinadas, como tiempo, coste y alcance [5]. Este campo de estudio y práctica se ha convertido en una disciplina clave dentro de las organizaciones, especialmente debido a la creciente complejidad y dinamismo en los entornos empresariales actuales, donde una gestión eficiente de proyectos es esencial para mantener la competitividad y asegurar el éxito en la ejecución estratégica [6].

La dirección de proyectos ha ido evolucionando a través de diferentes enfoques metodológicos. Inicialmente, dominaban metodologías tradicionales como la propuesta por el *Project Management Institute* (PMI), a través del PMBOK (*Project Management Body of Knowledge*), que ofrece un marco estructurado basado en procesos estandarizados para gestionar los proyectos [5]. Paralelamente, otras metodologías tradicionales, como el enfoque del International Project Management Association (IPMA), mediante su marco de competencias IPMA *Competence Baseline* (ICB), proporcionan un modelo centrado en competencias individuales, técnicas y contextuales del gerente de proyecto [7]. Asimismo, PRINCE2 (*Projects IN Controlled Environments*), desarrollada inicialmente en Reino Unido, propone una metodología basada en procesos claramente definidos y orientados al control riguroso del proyecto [8].

En las últimas décadas, respondiendo a la necesidad de una mayor flexibilidad ante entornos inciertos y cambiantes, han ganado relevancia las metodologías ágiles como SCRUM o KANBAN. Estas prácticas promueven la adaptación continua, la comunicación efectiva y la entrega iterativa e incremental de resultados, lo que permite a las organizaciones gestionar proyectos en contextos complejos e impredecibles [9,10]. Asimismo, emergen nuevos marcos metodológicos como PM², impulsado por la Comisión Europea, que integra elementos predictivos y ágiles para dar respuesta a las necesidades actuales de eficiencia, transparencia y gobernanza efectiva de los proyectos [11].

Ante este panorama metodológico, la transformación digital se ha convertido en un eje estratégico para muchas organizaciones en el ámbito de la gestión de proyectos. De hecho, metodologías como PMBOK o Agile ya han comenzado a integrar herramientas digitales para la planificación, el control de tareas o la colaboración en tiempo real. Sin embargo, la evolución digital no ha sido uniforme ni plenamente integrada en todos los marcos metodológicos, y en muchos casos aún persisten prácticas manuales, documentación en papel o procesos fragmentados.

La creciente complejidad y volumen de información, junto con la necesidad de mayor transparencia, trazabilidad y automatización, están acelerando la demanda de soluciones más robustas y descentralizadas. En especial, desafíos como la coordinación de equipos distribuidos, la gestión eficiente de recursos y el seguimiento en tiempo real del progreso del proyecto exigen tecnologías avanzadas capaces de garantizar la integridad, seguridad y accesibilidad de la información generada a lo largo del ciclo de vida del proyecto [12].

En respuesta a estos retos emergentes, la tecnología *blockchain* se posiciona como una herramienta prometedora que podría revolucionar la gestión de proyectos en términos de confianza, transparencia, seguridad y trazabilidad. *blockchain*, entendida como una tecnología descentralizada y distribuida basada en registros inmutables y consensuados, proporciona nuevas posibilidades para resolver cuestiones cruciales en la dirección moderna de proyectos, especialmente en entornos donde la confianza y la precisión en la gestión de la información son críticas [13].

Aunque todavía en fase inicial de adopción dentro del ámbito de la dirección de proyectos, *blockchain* anticipa beneficios significativos, especialmente en la mejora del control, auditoría y automatización de los procesos asociados a la gestión de proyectos. La próxima sección profundizará en el concepto y potencialidades de la Web3 y *blockchain*, explorando en detalle su aplicabilidad específica en contextos de dirección de proyectos.

1.2 Tecnologías Emergentes Aplicadas a la Gestión de Proyectos

La evolución tecnológica experimentada durante las últimas décadas ha dado lugar a un nuevo paradigma industrial y organizacional conocido como Industria 4.0. Este fenómeno, caracterizado por la integración digital avanzada y la automatización inteligente de procesos, ha transformado radicalmente la manera en que las organizaciones generan valor, gestionan sus recursos y ejecutan sus proyectos [14]. La digitalización se ha convertido en un factor clave no solo para mejorar la eficiencia y productividad operativa, sino también para incrementar la capacidad de innovación, adaptabilidad y toma de decisiones estratégicas dentro de los proyectos.

En este contexto, las tecnologías emergentes adquieren un rol central en la gestión moderna de proyectos, especialmente aquellas vinculadas al concepto de Web3, que marca una evolución significativa desde las plataformas digitales centralizadas hacia ecosistemas descentralizados basados en *blockchain* [13]. La tecnología *blockchain*, por sus características intrínsecas de descentralización, transparencia e inmutabilidad, facilita la resolución de desafíos tradicionales en la gestión de proyectos, especialmente en áreas sensibles como la trazabilidad, la gobernanza transparente y la automatización confiable de procesos.

Entre los desarrollos más relevantes asociados a *blockchain* para la gestión de proyectos se encuentran los contratos inteligentes (*smart contract*s), que permiten automatizar acuerdos y transacciones sin necesidad de intermediarios, aumentando significativamente la eficiencia operativa y la confiabilidad en la ejecución contractual [15,16]. Además, la diversidad de redes *blockchain* existentes, como Ethereum, Hyperledger Fabric, Binance Smart Chain, Solana o Polkadot, proporciona diferentes marcos tecnológicos que pueden adaptarse específicamente a las necesidades y características particulares de cada proyecto.

En la presente sección se analizarán de manera detallada estas tecnologías, destacando especialmente cómo la Web3, *blockchain* y los contratos inteligentes están revolucionando la gestión de proyectos, mejorando la eficiencia, transparencia y confiabilidad en procesos clave, y cómo diversas redes *blockchain* pueden aplicarse estratégicamente para abordar retos específicos dentro del ámbito proyectual.

1.2.1. **Web3: Un Nuevo Paradigma para la Gestión Descentralizada de Proyectos**

La evolución tecnológica y el avance hacia la Industria 4.0 han supuesto un profundo cambio en los modelos organizativos, potenciando entornos altamente digitalizados, interconectados y descentralizados. En este contexto, el concepto de Web3 emerge como un nuevo paradigma de internet descentralizado que, apoyado en tecnologías como *blockchain*, *smart contract*s e identidad digital descentralizada, busca transformar radicalmente la forma en que usuarios e instituciones interactúan en entornos digitales [17]. Si la Web 2.0 se centró en plataformas centralizadas dominadas por grandes corporaciones tecnológicas que gestionan y explotan la información generada por los usuarios, Web3 plantea un modelo opuesto donde la propiedad, la gestión de la información y la identidad digital residen directamente en manos de los usuarios, reduciendo drásticamente la necesidad de intermediarios y aumentando la transparencia y la confianza [18].

Uno de los elementos más disruptivos que Web3 introduce es la gestión descentralizada de identidades digitales (DID). Esta tecnología permite a los usuarios poseer y controlar plenamente sus credenciales digitales, eliminando la necesidad de entidades intermediarias y asegurando una interoperabilidad segura y transparente en múltiples plataformas y aplicaciones [17]. En proyectos empresariales y organizacionales, esta capacidad facilita enormemente la colaboración entre actores distribuidos, simplificando procesos como la autenticación, validación y auditoría de identidades y roles dentro de un equipo de trabajo o consorcio de empresas.

En esta línea, Web3 también fomenta el surgimiento de modelos económicos descentralizados basados en activos digitales como tokens fungibles (criptomonedas) y tokens no fungibles (NFTs). Estos activos facilitan nuevas formas de propiedad digital, comercio virtual y gestión económica dentro del ecosistema digital. En particular, los NFTs permiten tokenizar cualquier tipo de activo digital o físico, facilitando una gestión eficiente, segura y transparente de los derechos de propiedad intelectual y activos tangibles o intangibles dentro de los proyectos [19]. Así, la implementación de Web3 en la dirección de proyectos abre oportunidades para gestionar activos, recursos y documentación de manera descentralizada y automatizada, favoreciendo la reducción de costes operativos y aumentando la eficiencia en los procesos.

Pese a estas ventajas potenciales, la adopción práctica de Web3 presenta desafíos importantes, principalmente relacionados con la volatilidad económica inherente a las criptomonedas utilizadas en este entorno, como Ethereum, que impacta en la estabilidad económica y previsibilidad presupuestaria necesarias en la gestión de proyectos [18]. La alta volatilidad y el comportamiento especulativo de estos activos pueden representar riesgos significativos si no se gestionan adecuadamente. Sin embargo, es precisamente aquí donde las *stablecoins* o monedas estables, criptomonedas diseñadas para mantener un valor estable respecto a monedas fiduciarias tradicionales, pueden ser más adecuadas como medio de pago y gestión financiera en el marco de proyectos.

Asimismo, otro reto clave reside en la interoperabilidad y la gobernanza descentralizada de la información generada y compartida dentro de los proyectos, donde *blockchain* juega un papel fundamental. *blockchain*, al ser una tecnología distribuida basada en registros digitales inmutables y criptográficamente asegurados, permite establecer mecanismos fiables de gestión descentralizada y colaborativa, facilitando auditorías, control de cambios y trazabilidad de procesos [13].

Por tanto, desde un enfoque práctico y aplicado a la gestión de proyectos, Web3 no sólo implica un cambio tecnológico, sino también una transformación cultural y organizacional. Su implementación efectiva requiere repensar los esquemas tradicionales de control, gestión de recursos e interacción con actores externos e internos del proyecto. En consecuencia, resulta fundamental comprender tanto las posibilidades tecnológicas como los desafíos económicos y organizacionales inherentes a su

implementación, para asegurar una transición efectiva y segura hacia la adopción del paradigma Web3 en la dirección de proyectos.

En las siguientes subsecciones, se abordarán específicamente las tecnologías clave dentro de Web3, como *blockchain*, *smart contract*s, y algunas de las redes *blockchain* más relevantes, profundizando en su aplicabilidad práctica y potencial transformador en la gestión contemporánea de proyectos.

1.2.2. *Blockchain*: Fundamentos, Aplicabilidad y Retos para la Gestión de Proyectos

La tecnología *blockchain* ha surgido como una innovación disruptiva con potencial para transformar radicalmente numerosos sectores económicos y organizacionales. En esencia, *blockchain* es una estructura de datos descentralizada que almacena registros en bloques enlazados cronológicamente, asegurando la integridad, transparencia e inmutabilidad de la información contenida en ellos mediante técnicas criptográficas y algoritmos de consenso [20,21]. Esta tecnología nació como soporte fundamental para criptomonedas como Bitcoin, pero pronto trascendió su origen para aplicarse en diversos campos que incluyen la gestión de la cadena de suministro, servicios financieros, salud, identidad digital y procesos industriales [22].

Blockchain puede clasificarse principalmente en tres tipos según sus características de acceso y control: *blockchain* pública, privada y de consorcio. La *blockchain* pública, como Bitcoin o Ethereum, permite el acceso y participación libre, basándose en mecanismos abiertos de consenso como la prueba de trabajo (PoW). La *blockchain* privada está restringida a un grupo limitado de participantes autorizados, proporcionando mayores niveles de privacidad, eficiencia y control en las transacciones. Finalmente, las *blockchain*s de consorcio o híbridas se caracterizan por estar gobernadas por múltiples entidades autorizadas que colaboran bajo acuerdos previamente establecidos, facilitando la cooperación empresarial interorganizacional [23].

Una de las principales características que hacen atractiva esta tecnología para la gestión de proyectos es su capacidad de ofrecer transparencia y trazabilidad en procesos complejos, permitiendo la monitorización en tiempo real del estado y avance de proyectos. La tecnología *blockchain* facilita además la creación de redes descentralizadas donde todos los actores involucrados pueden confiar en la información compartida sin depender de intermediarios o autoridades centralizadas. Esto contribuye a reducir significativamente riesgos operativos como fraudes, errores humanos o manipulaciones, asegurando la integridad de las transacciones, documentos y comunicaciones dentro del ciclo de vida del proyecto [20].

En la gestión de proyectos, *blockchain* proporciona una infraestructura tecnológica adecuada para implementar contratos inteligentes (*smart contract*s). Estos son acuerdos autoejecutables escritos en código informático, capaces de ejecutar automáticamente cláusulas contractuales al cumplirse condiciones predefinidas, facilitando una automatización confiable y eficiente de procesos como pagos, aprobaciones, y validaciones sin intermediación humana. La incorporación de contratos inteligentes permite reducir costos operativos, minimizar riesgos de incumplimiento contractual y garantizar que todas las partes del proyecto cumplen estrictamente con sus obligaciones contractuales [20,23].

Otra dimensión significativa de *blockchain* en el ámbito de la gestión de proyectos se vincula a la seguridad y privacidad de los datos. Aunque *blockchain* es reconocida por su capacidad para asegurar la integridad de la información, también presenta retos importantes relacionados con el cumplimiento de regulaciones estrictas como el Reglamento General de Protección de Datos (GDPR) [24]. La

inmutabilidad intrínseca de *blockchain* genera conflictos con derechos digitales, como el derecho al olvido o la supresión de datos personales [25]. Este desafío ha impulsado la investigación en nuevas soluciones técnicas y modelos híbridos que armonicen la transparencia y seguridad proporcionadas por *blockchain* con las normativas legales vigentes.

Blockchain también se está posicionando como una tecnología clave para alcanzar objetivos de desarrollo sostenible (ODS) [26] en los sistemas productivos y proyectos industriales. Su capacidad para mejorar la transparencia y eficiencia en procesos industriales promueve prácticas sostenibles, facilitando la reducción del desperdicio de recursos, optimizando la trazabilidad en cadenas de suministro y asegurando el cumplimiento regulatorio y normativo [22].

No obstante, a pesar de las ventajas mencionadas, la adopción efectiva de *blockchain* en la gestión de proyectos presenta varios retos técnicos, económicos y organizacionales. Las limitaciones relacionadas con la escalabilidad, el consumo energético, y la complejidad técnica del despliegue inicial constituyen barreras importantes. Además, se requiere una transición cultural y organizacional significativa para que las empresas adopten plenamente esta tecnología, lo que implica cambios profundos en las prácticas tradicionales de gestión y control [20,23].

En breve síntesis, *blockchain* ofrece un potencial considerable para transformar positivamente la gestión de proyectos, garantizando mayor transparencia, seguridad, automatización y eficiencia operativa. Sin embargo, la efectiva implementación de esta tecnología requiere una cuidadosa consideración de sus limitaciones técnicas y regulatorias, así como la preparación adecuada de las organizaciones para adoptar y gestionar estos cambios profundos en su estructura y cultura operativa.

1.2.3. Redes *Blockchain* Existentes: Clasificación, Características y Aplicabilidad en la Gestión de Proyectos

La tecnología *blockchain* se materializa mediante diversas redes o plataformas, las cuales condicionan la manera específica en la que se implementan sus principios fundamentales como la descentralización, el consenso distribuido, la privacidad y la seguridad criptográfica [20,23]. La elección de una red *blockchain* adecuada en proyectos depende directamente del tipo de aplicación, las necesidades relacionadas con privacidad, transparencia y escalabilidad, así como del nivel de acceso y participación deseados.

En esta sección se detallan exhaustivamente diferentes tipos de redes *blockchain*, agrupadas en tres categorías principales según su acceso y gobernanza: públicas, privadas y de consorcio o híbridas. Dentro de estas clasificaciones se integran tanto las redes más populares como aquellas menos conocidas pero igualmente relevantes.

1.2.3.1 *Blockchain* Pública

Las redes *blockchain* públicas son abiertas, descentralizadas y permiten que cualquier usuario participe libremente sin restricciones previas. Operan mediante algoritmos de consenso abiertos que buscan garantizar la seguridad, transparencia y resistencia a la censura [20]. Algunas de las redes más destacadas y sus características principales son:

- **Bitcoin**: es la primera *blockchain* pública implementada, pionera en criptomonedas y tecnología *blockchain*. Utiliza un mecanismo de consenso basado en Prueba de Trabajo (*Proof-of-Work*, PoW), lo que garantiza una seguridad excepcional frente a ataques, pero

implica una elevada demanda energética y limitaciones significativas en escalabilidad. Su principal aplicación sigue siendo el almacenamiento de valor y transacciones monetarias seguras [27].

- **Ethereum**: *blockchain* ampliamente reconocida por introducir contratos inteligentes y aplicaciones descentralizadas (DApps). Destaca por su capacidad programable mediante la *Ethereum Virtual Machine* (EVM), que permite crear aplicaciones avanzadas en sectores financieros (DeFi), juegos (GameFi), NFTs y gobernanza descentralizada (DAO). Ethereum recientemente migró a Ethereum 2.0 adoptando la Prueba de Participación (*Proof-of-Stake*, PoS), mejorando drásticamente su eficiencia energética y escalabilidad, posicionándose como la plataforma estándar para contratos inteligentes [16,28].

- **Solana**: red *blockchain* pública orientada al alto rendimiento, capaz de procesar hasta 50.000 transacciones por segundo. Su mecanismo híbrido combina PoS con la innovadora Prueba de Historia (*Proof-of-History*, PoH), logrando tiempos de confirmación extremadamente rápidos y tarifas muy reducidas. Solana es especialmente adecuada para aplicaciones descentralizadas que requieren alta velocidad, como finanzas descentralizadas avanzadas (DeFi), mercados de tokens no fungibles (NFTs), juegos *blockchain* y plataformas Web3 con experiencia de usuario optimizada [29].

- **Cardano**: *blockchain* pública diseñada con enfoque académico y científico, conocida por su rigor en seguridad y estabilidad técnica. Utiliza Ouroboros, un algoritmo de consenso PoS altamente estudiado y auditado. Cardano es idónea para proyectos donde la precisión y conformidad regulatoria son esenciales, como soluciones financieras institucionales, gestión de identidad digital, gobernanza descentralizada y trazabilidad de productos en cadenas de suministro críticas [30].

- **EOSIO**: plataforma *blockchain* optimizada para aplicaciones de alto rendimiento transaccional mediante el consenso de Prueba Delegada de Participación (*Delegated Proof-of-Stake*, DPoS). EOSIO destaca por su rapidez de procesamiento y escalabilidad horizontal. Resulta especialmente atractiva para plataformas sociales descentralizadas, juegos interactivos, servicios multimedia distribuidos y aplicaciones con una amplia base de usuarios activos simultáneos [31].

- **Binance Smart Chain (BSC)**: *blockchain* pública compatible con contratos inteligentes de Ethereum (EVM-compatible), pero diseñada para transacciones más rápidas y económicas mediante el mecanismo de consenso PoS Autoritativo (*Proof-of-Staked Authority*, PoSA). BSC es particularmente popular en el ecosistema financiero descentralizado (DeFi) y de NFTs, resultando atractiva para proyectos que buscan una alternativa económica a Ethereum sin perder la compatibilidad tecnológica y funcional [32].

- **Sui**: *blockchain* emergente enfocada en mejorar considerablemente la experiencia del usuario mediante una arquitectura innovadora basada en PoS Delegado. Se orienta hacia aplicaciones Web3, metaversos y juegos *blockchain* de última generación, ofreciendo rendimiento excepcional, bajas latencias y facilidad de desarrollo. Su capacidad de procesamiento paralelo la hace atractiva para proyectos que requieren simultaneidad y alta escalabilidad en transacciones complejas [33].

- **Avalanche**: plataforma *blockchain* pública altamente escalable y rápida, destacada por su innovador consenso Avalanche (PoS), que proporciona tiempos de confirmación inferiores a un segundo con alta seguridad y descentralización. Compatible con Ethereum (EVM), facilita la migración de aplicaciones existentes. Es ideal para DeFi, gestión descentralizada de activos

digitales, plataformas de inversión y aplicaciones empresariales que requieren seguridad, rapidez y escalabilidad robusta [34].

- **Polygon**: solución de escalabilidad que opera como una *blockchain* pública paralela y complementaria a Ethereum. Compatible con contratos inteligentes (EVM-compatible), Polygon reduce significativamente los costes de transacción y mejora el rendimiento. Utiliza un mecanismo híbrido de consenso PoS, siendo ideal para aplicaciones descentralizadas (DApps), DeFi, juegos *blockchain* y plataformas NFT que necesitan alta velocidad y bajos costos operativos, aprovechando al máximo el ecosistema existente de Ethereum [35].

El gráfico elaborado en Python [36] que se presenta a continuación (Figura 1) muestra una evaluación comparativa, de las redes *blockchain* públicas recién comentadas, en función de seis características clave: escalabilidad, coste de transacción, compatibilidad con EVM, rendimiento/latencia, descentralización y ecosistema/casos de uso. Esta visualización permite identificar de forma clara las fortalezas y debilidades relativas de cada red.

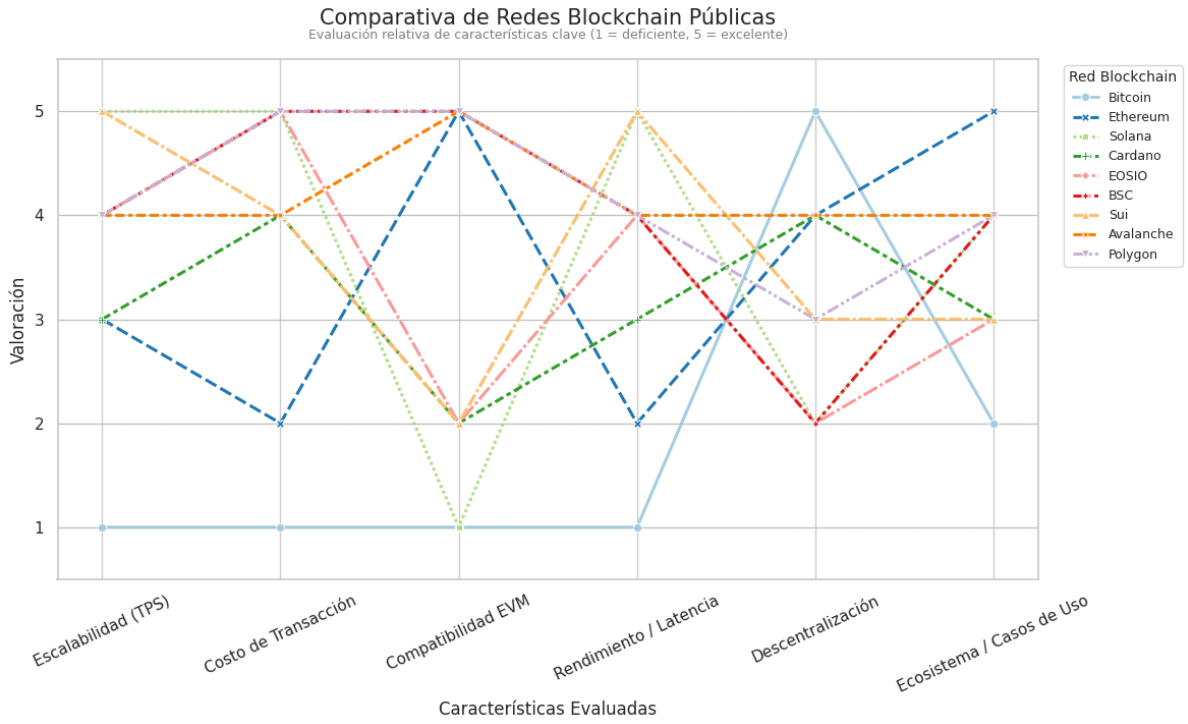

Figura 1. Comparativa de redes *blockchain* públicas. Elaboración propia con Python [36].

1.2.3.2 *Blockchain* Privada

Las redes *blockchain* privadas permiten el acceso únicamente a un grupo selecto y autorizado de participantes, ofreciendo control absoluto sobre el acceso, privacidad avanzada y alta eficiencia operativa en entornos empresariales o regulados [23]. Estas redes son particularmente adecuadas cuando las organizaciones requieren alta confidencialidad, personalización y cumplimiento normativo estricto. Entre las principales plataformas destacan:

- **Hyperledger Fabric**: plataforma *blockchain* modular desarrollada por la Linux *Foundation* [37]. Destaca por su flexibilidad en la configuración de nodos, roles y algoritmos de consenso personalizados según las necesidades empresariales. Fabric proporciona mecanismos

avanzados de privacidad a través de canales privados y transacciones confidenciales, convirtiéndola en ideal para sectores sensibles como el financiero, sanitario y gestión de cadenas de suministro complejas donde la privacidad y control interno son cruciales [38].

- **Corda (R3)**: diseñada específicamente para aplicaciones empresariales y financieras con un énfasis absoluto en la privacidad transaccional. Su mecanismo de consenso, basado en acuerdos entre partes directamente involucradas, garantiza que solo los nodos participantes de una transacción puedan acceder a sus detalles. Corda es especialmente útil en mercados regulados como banca, seguros, comercio internacional y plataformas de intercambio de activos digitales, donde la privacidad selectiva y el cumplimiento normativo son requisitos críticos [39].

- **Quorum**: plataforma *blockchain* privada basada en Ethereum, inicialmente desarrollada por JP Morgan, adaptada específicamente para entornos financieros regulados. Quorum combina la potencia y flexibilidad de los contratos inteligentes de Ethereum con características adicionales de privacidad avanzada mediante protocolos como "*Zero-Knowledge Proofs*" y privacidad por consenso restringido. Se caracteriza por su capacidad para ejecutar transacciones rápidas, confidenciales y compatibles con entornos bancarios y financieros estrictamente regulados [40].

Para visualizar de forma comparativa las características clave de las principales redes *blockchain* privadas, se ha elaborado un gráfico que recoge una evaluación relativa de plataformas como Hyperledger Fabric, Corda y Quorum. En la Figura 2 se representan estas valoraciones según criterios como privacidad, configurabilidad, rendimiento, madurez empresarial, cumplimiento normativo y compatibilidad con Ethereum (centrada en contratos inteligentes), proporcionando así un soporte visual al análisis desarrollado en este apartado.

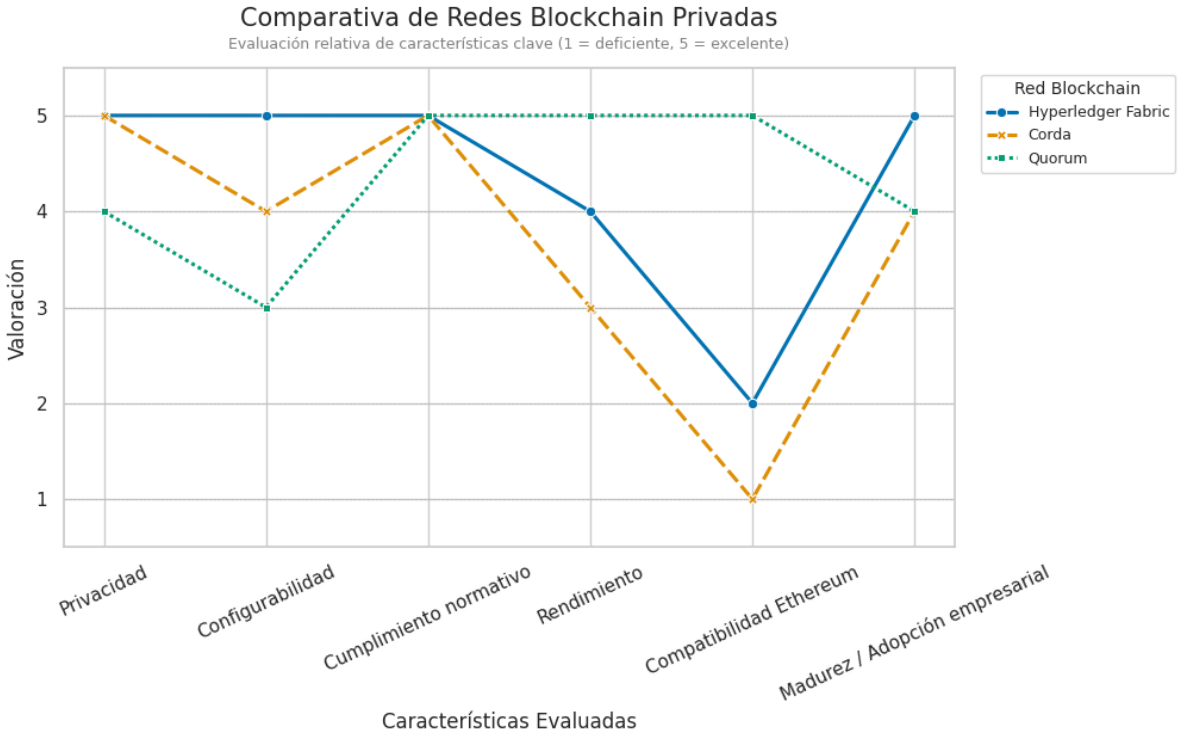

Figura 2. Comparativa de redes *blockchain* privadas. Elaboración propia con Python [36].

1.2.3.3 *Blockchain* de Consorcio o Híbrida

Las redes *blockchain* híbridas o de consorcio se caracterizan por tener un acceso semi-descentralizado, siendo gestionadas cooperativamente por varias entidades autorizadas. Este modelo proporciona un balance estratégico entre transparencia pública y privacidad controlada, siendo adecuado para contextos colaborativos en los que varias organizaciones interactúan bajo normas compartidas claramente definidas [20]. Algunas plataformas relevantes son:

- **Hyperledger Besu**: *blockchain* de consorcio basada en Ethereum, diseñada específicamente para entornos empresariales con necesidades mixtas de transparencia y privacidad. Besu permite la creación de redes privadas con transacciones confidenciales dentro de una estructura Ethereum-compatible, facilitando la interoperabilidad con sistemas *blockchain* públicos. Es muy utilizada en consorcios financieros, colaboración interempresarial en cadenas de suministro y redes de gestión de activos compartidos, gracias a su capacidad de integrar contratos inteligentes con privacidad ajustable [41].

- **Ripple**: plataforma *blockchain* híbrida especializada en facilitar pagos transfronterizos e intercambio de divisas rápidas y económicas. Su mecanismo propio de consenso (Ripple *Protocol Consensus Algorithm*, RPCA) se basa en nodos validadores seleccionados previamente, proporcionando velocidad transaccional elevada y bajos costes operativos. Ripple es especialmente popular entre instituciones financieras globales que requieren eficiencia, rapidez y cumplimiento normativo en sus operaciones de transferencia de fondos internacionales [42].

- **Hyperledger Sawtooth**: *blockchain* híbrida con un mecanismo innovador de consenso denominado Prueba de Tiempo Transcurrido (*Proof-of-Elapsed-Time*, PoET), desarrollado originalmente por Intel. Este enfoque proporciona escalabilidad robusta y seguridad en redes empresariales complejas, especialmente adecuado para aplicaciones industriales, IoT empresarial y entornos colaborativos interorganizacionales donde múltiples actores requieren gobernanza compartida, pero manteniendo ciertos grados de privacidad operativa [43].

- **Polkadot**: plataforma *blockchain* híbrida enfocada en la interoperabilidad multi*blockchain*, facilitando la comunicación e intercambio de información y valor entre múltiples *blockchain*s, tanto públicas como privadas. Su arquitectura modular permite a las redes privadas operar de forma aislada, pero conectadas mediante un protocolo centralizado, garantizando escalabilidad, seguridad e interoperabilidad. Polkadot es adecuada para proyectos descentralizados de gran escala que necesitan integrarse fluidamente con múltiples ecosistemas *blockchain* [44].

- **VeChain**: *blockchain* híbrida especializada en gestión de cadenas de suministro y trazabilidad de productos mediante tecnologías IoT integradas. VeChain permite seguimiento transparente y verificable de productos físicos en tiempo real, asegurando su autenticidad y calidad mediante sensores IoT y etiquetas inteligentes. Su modelo híbrido permite el acceso controlado a la información, balanceando transparencia pública para consumidores y privacidad interna para empresas, lo que la hace idónea en sectores como alimentación, lujo, moda y logística avanzada [45].

- **IOTA**: Plataforma híbrida que introduce una estructura innovadora sin bloques denominada Tangle, basada en un Grafo Acíclico Dirigido (DAG). IOTA es ideal para entornos industriales 4.0, gestión de dispositivos IoT y aplicaciones que requieren alta velocidad transaccional sin comisiones, garantizando escalabilidad prácticamente ilimitada. Su estructura descentralizada

híbrida permite una gestión eficiente y segura de datos generados por sensores, máquinas autónomas y entornos industriales distribuidos [46].

Con el fin de comparar visualmente las principales plataformas *blockchain* de tipo consorcio o híbridas, se ha elaborado una representación gráfica que sintetiza su desempeño en distintos aspectos técnicos y funcionales. La Figura 3 presenta una evaluación relativa de soluciones como Hyperledger Besu, Ripple, VeChain, IOTA o Polkadot, según criterios como escalabilidad, interoperabilidad, privacidad, gobernanza compartida y compatibilidad con entornos empresariales e industriales, así como con Ethereum.

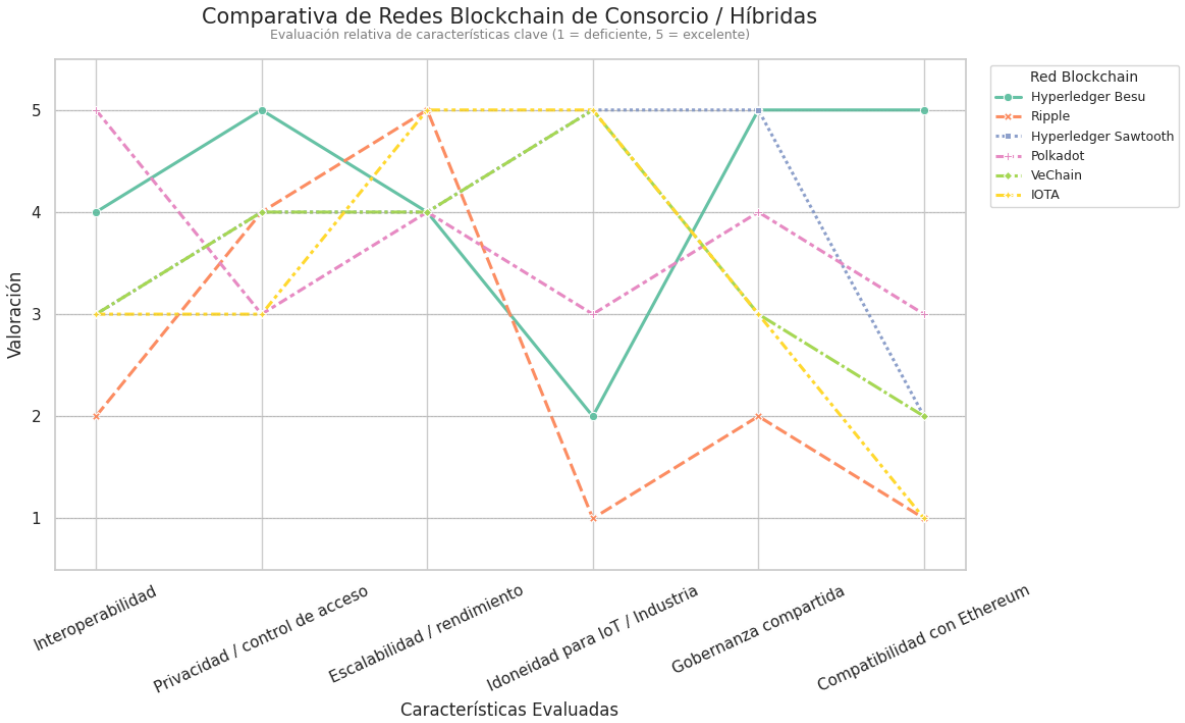

Figura 3. Comparativa de redes *blockchain* de consorcio/híbridas. Elaboración propia con Python [36].

La elección adecuada de una red *blockchain* depende directamente de factores clave como los objetivos del proyecto, las necesidades de privacidad, el rendimiento requerido y la interoperabilidad deseada. En contextos financieros descentralizados (DeFi), plataformas públicas como Ethereum, BSC, Avalanche, Polygon o Solana ofrecen ventajas claras por su capacidad de contratos inteligentes y bajos costes. Para aplicaciones empresariales reguladas, la privacidad avanzada y personalización de redes privadas como Hyperledger Fabric, Corda o Quorum son más adecuadas. En proyectos industriales o logísticos que integran IoT e Industria 4.0, redes híbridas como Hyperledger Sawtooth, VeChain o IOTA destacan por escalabilidad y conexión directa con dispositivos físicos. Polkadot e Hyperledger Besu son ideales cuando la interoperabilidad entre múltiples *blockchain*s es crítica, mientras que EOSIO o Sui son preferibles para aplicaciones Web3, juegos y metaversos que requieren alto rendimiento y usabilidad. En definitiva, una selección informada basada en las características específicas de cada plataforma asegurará el éxito y eficiencia en la implementación *blockchain* en cualquier proyecto.

Para facilitar la comprensión de las principales diferencias entre los distintos tipos de redes *blockchain* (públicas, privadas e híbridas/consorcio), se ha elaborado una representación comparativa que resume sus características clave. La Figura 4 muestra una visualización en forma de radar que evalúa

dimensiones como descentralización, privacidad, interoperabilidad, gobernanza compartida, escalabilidad, transparencia y rendimiento. Esta representación ayuda a contrastar visualmente sus fortalezas y limitaciones, sirviendo de apoyo al análisis narrativo previo sobre su aplicabilidad según el contexto de uso.

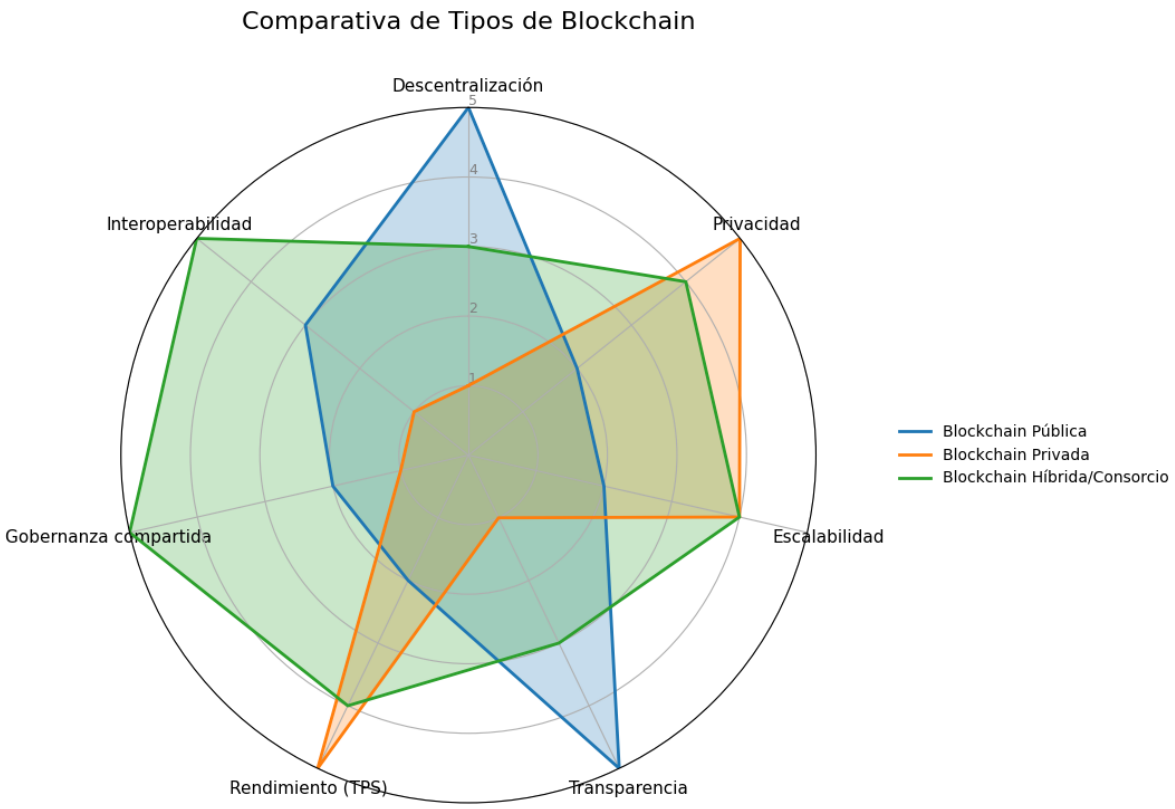

Figura 4. Comparativas de tipos de *blockchain*. Elaboración propia con Python [36].

1.2.4. Contratos Inteligentes: Concepto, Lenguajes de Programación y Aplicabilidad en Gestión de Proyectos

Los contratos inteligentes (*smart contract*s) son programas informáticos que se almacenan y ejecutan automáticamente en plataformas *blockchain*, permitiendo llevar a cabo acuerdos y transacciones entre diferentes actores sin necesidad de intermediarios centralizados o de confianza [47]. Este término fue introducido inicialmente por Nick Szabo, quien concibió estos contratos como protocolos informáticos capaces de automatizar cláusulas contractuales predefinidas, asegurando una ejecución transparente, segura e incorruptible [48].

En términos operativos, un contrato inteligente consiste en un código informático alojado en *blockchain*, que automáticamente verifica y ejecuta las cláusulas previamente acordadas por las partes involucradas cuando se cumplen ciertas condiciones específicas. Los resultados de esta ejecución son registrados de forma inmutable en la red *blockchain* correspondiente, garantizando trazabilidad y transparencia absoluta en las transacciones realizadas [49].

El ciclo de vida de un contrato inteligente contempla cuatro fases principales: creación (codificación de condiciones contractuales), despliegue (puesta en producción en la *blockchain* elegida), ejecución

(activación automática ante cumplimiento de condiciones) y finalización (registro inmutable del resultado en *blockchain*) [47].

Existen diversos lenguajes especializados en la programación de contratos inteligentes, algunos diseñados específicamente para este propósito y otros adaptados desde entornos tradicionales.
A continuación, se presentan los lenguajes más destacados junto con las plataformas *blockchain* que los utilizan habitualmente:

- **Solidity**: lenguaje de alto nivel más popular y extendido para contratos inteligentes. Su sintaxis está influenciada por lenguajes como JavaScript y C++. Solidity se ejecuta principalmente en Ethereum y plataformas compatibles como Binance Smart Chain (BSC), Polygon o Avalanche [50].

- **Vyper**: lenguaje enfocado en seguridad, simplicidad y legibilidad, también diseñado para Ethereum y redes compatibles con la máquina virtual Ethereum (EVM). Vyper está diseñado para minimizar errores comunes y vulnerabilidades en el desarrollo de contratos inteligentes [48].

- **Rust**: lenguaje de propósito general con alto rendimiento y seguridad. Ha ganado popularidad recientemente en redes *blockchain* como Solana, Near y Polkadot, principalmente por sus capacidades avanzadas de gestión de memoria y optimización para alta eficiencia y seguridad crítica [48].

- **Kotlin y Java**: lenguajes ampliamente utilizados en entornos empresariales, especialmente en la *blockchain* Corda (R3), donde se combinan cláusulas legales con ejecución técnica. Su empleo facilita la integración con sistemas heredados y contextos financieros o legales altamente regulados [48].

- **Go (Golang)**: desarrollado por Google con un fuerte énfasis en rendimiento, concurrencia y sencillez. Es utilizado frecuentemente en plataformas empresariales como Hyperledger Fabric y otras plataformas *blockchain* privadas o híbridas por su escalabilidad y capacidad de integración en infraestructuras IT existentes [48].

- **C++ y WebAssembly (WASM)**: utilizados principalmente en EOSIO debido a su velocidad y eficiencia en ejecución. La adopción de estos lenguajes permite manejar gran cantidad de transacciones simultáneas, resultando idóneo para aplicaciones intensivas como juegos, plataformas multimedia y aplicaciones interactivas descentralizadas [48].

- **Move**: lenguaje reciente enfocado en seguridad y protección de activos digitales, usado en redes emergentes como Sui y Aptos. Destaca por garantizar formalmente que ciertas propiedades críticas sean siempre preservadas, especialmente útil en aplicaciones financieras y gestión avanzada de activos digitales [33].

Con el objetivo de seleccionar un lenguaje de programación adecuado para la implementación de contratos inteligentes, se ha realizado un análisis comparativo de las principales opciones disponibles en el ecosistema *blockchain*. La Figura 5 recoge una evaluación visual de distintos lenguajes en función de criterios como eficiencia, legibilidad, curva de aprendizaje, seguridad, uso típico (es decir, que tan comúnmente se utiliza cada lenguaje), compatibilidad *blockchain* (esto es, la capacidad del lenguaje para integrarse con redes *blockchain* concretas) y madurez del ecosistema. Esta representación gráfica permite contrastar rápidamente las fortalezas y debilidades de cada lenguaje, y justifica la posterior elección de Solidity como el más adecuado para este proyecto.

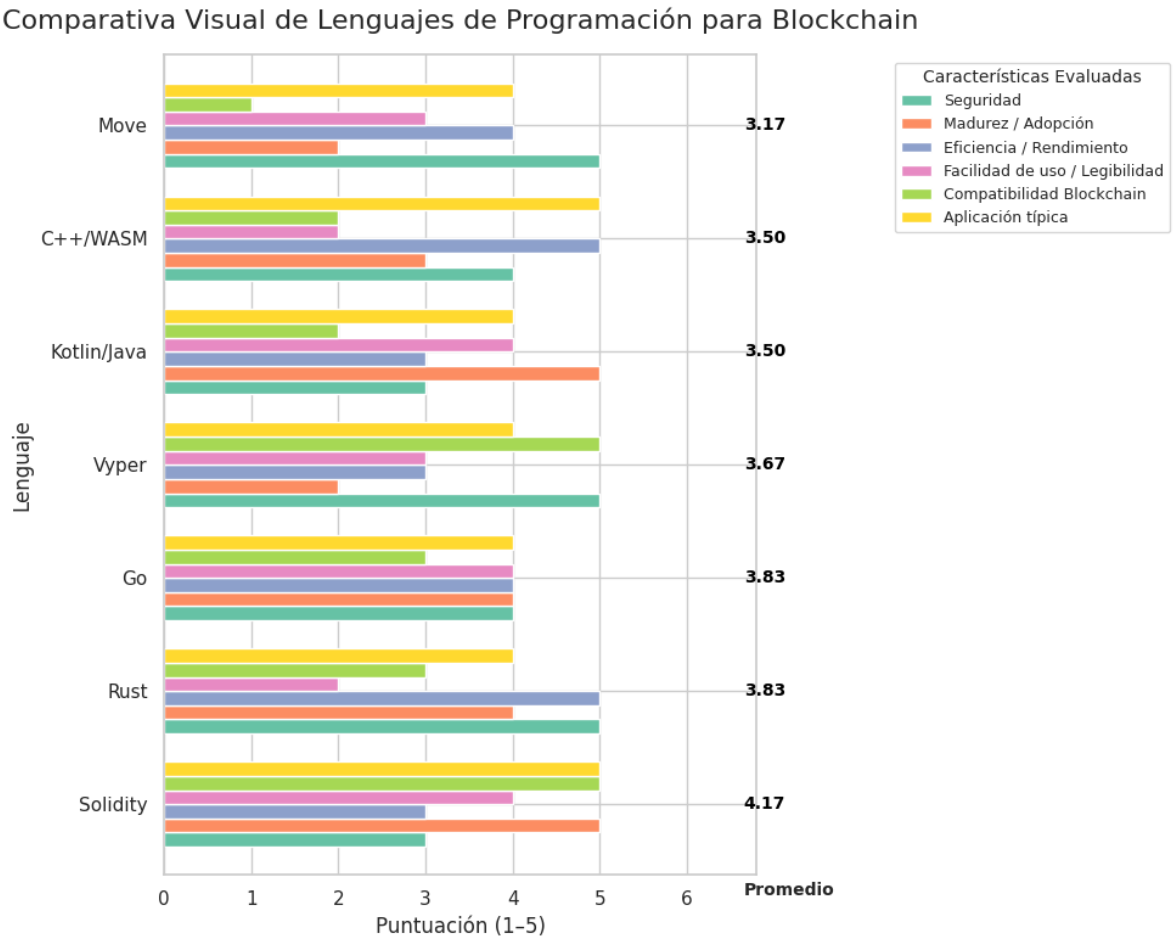

Figura 5. Comparativa de lenguajes de programación para *blockchain*. Elaboración propia con Python [36].

La implementación de contratos inteligentes en la gestión de proyectos ofrece múltiples beneficios específicos:

- **Automatización contractual**: los contratos inteligentes ejecutan automáticamente cláusulas específicas cuando se cumplen ciertas condiciones, eliminando la necesidad de intervención humana en procesos administrativos repetitivos y reduciendo significativamente tiempos y costes operativos [47].

- **Incremento de la transparencia y auditabilidad**: al almacenar toda la información relevante y acciones ejecutadas en *blockchain*, se asegura una transparencia total en la ejecución contractual, permitiendo la auditoría continua por todas las partes involucradas [48].

- **Mayor seguridad y reducción del riesgo de fraude**: debido a su naturaleza descentralizada e inmutable, los contratos inteligentes proporcionan alta seguridad frente a manipulaciones o fraudes, minimizando riesgos contractuales y conflictos entre las partes involucradas [50].

- **Eficiencia económica y reducción de costos operativos**: la reducción en tiempos de ejecución contractual y eliminación de intermediarios tradicionales conlleva un ahorro considerable en costes administrativos y legales, aumentando la eficiencia económica global del proyecto [49].

Sin embargo, el uso de contratos inteligentes en proyectos también presenta algunas limitaciones específicas:

- **Complejidad técnica y vulnerabilidades potenciales**: la correcta codificación de contratos inteligentes requiere competencias avanzadas en programación segura y auditoría informática. Errores en el código pueden ocasionar pérdidas económicas significativas, como se ha evidenciado en casos conocidos como el ataque DAO en Ethereum [50].

- **Problemas regulatorios y reconocimiento legal**: la mayoría de las jurisdicciones aún carecen de marcos legales claros para el uso de contratos inteligentes, generando incertidumbre jurídica en disputas contractuales y limitando su aceptación en ciertos sectores muy regulados [49].

- **Irreversibilidad y rigidez en la ejecución contractual**: una vez desplegado un contrato inteligente, su lógica y ejecución no pueden modificarse fácilmente. Esto implica que errores en condiciones o cláusulas pueden tener consecuencias permanentes y significativas para el proyecto [47].

- **Limitaciones en la escalabilidad y rendimiento**: muchas plataformas *blockchain* presentan limitaciones en términos de capacidad de procesamiento, provocando congestión y altos costos de ejecución cuando el proyecto implica grandes volúmenes transaccionales simultáneos [48].

En definitiva, los contratos inteligentes constituyen una herramienta potencialmente transformadora para la gestión eficiente, transparente y segura de proyectos, aunque su implementación exitosa requiere una evaluación cuidadosa de sus ventajas frente a las limitaciones técnicas y jurídicas actuales.

1.3 Dirección de Proyectos y *Blockchain*

La gestión de proyectos se ha desarrollado históricamente mediante diferentes metodologías que buscan estructurar, planificar y controlar los recursos y actividades necesarios para alcanzar objetivos específicos. Cada una de estas metodologías (tradicionales, ágiles o híbridas) responde a contextos, sectores y niveles de complejidad distintos, y ha evolucionado para adaptarse a entornos organizacionales cada vez más dinámicos y digitalizados.

En los últimos años, la tecnología *blockchain* ha comenzado a posicionarse como un habilitador potencial en la gestión de proyectos, ofreciendo mejoras en áreas clave como la trazabilidad, la transparencia, la automatización contractual y la seguridad de la información. Sin embargo, su grado de adopción varía significativamente entre metodologías, y la literatura académica aún se encuentra en una etapa inicial respecto a su integración formal en marcos estructurados de dirección de proyectos.

En esta sección se analizarán las principales metodologías de gestión de proyectos (como PMBOK, PRINCE2, Agile, Lean, ICB y PM²) y se examinará si existen propuestas, estudios o aplicaciones reales que vinculen cada una de ellas con el uso de tecnología *blockchain*, con el objetivo de identificar oportunidades y brechas actuales en la investigación y la práctica profesional.

1.3.1. **Metodologías en Cascada (*Waterfall*)**

La metodología *waterfall*, también conocida como desarrollo en cascada, es un modelo de gestión de proyectos que sigue un enfoque lineal y secuencial. Este modelo fue ampliamente adoptado en la

década de 1970 y ha sido utilizado principalmente en industrias como la construcción, la ingeniería de software y la infraestructura [51]. Su estructura jerárquica permite definir un conjunto de fases rígidas que deben completarse en un orden específico antes de avanzar a la siguiente etapa del proyecto.

El modelo *waterfall* se compone de varias fases bien definidas:

- **Recolección de requisitos**: se identifican y documentan todos los requisitos del proyecto antes de comenzar su desarrollo.
- **Diseño**: se planifica detalladamente cómo se desarrollará el proyecto, estableciendo la arquitectura y las especificaciones técnicas.
- **Implementación**: se ejecuta el desarrollo o construcción de acuerdo con el diseño definido.
- **Verificación y Validación (V&V)**: se realizan pruebas para garantizar que el producto final cumple con los requisitos establecidos.
- **Despliegue y mantenimiento**: una vez finalizado el proyecto, se realizan ajustes y mantenimiento según sea necesario.

Atendiendo ahora a la propuesta de otros autores sobre como implementar *blockchain* en esta metodología, Will Serrano [52] propone un sistema de V&V basado en *blockchain* y contratos inteligentes que permite la autenticación y certificación de datos a lo largo de las diferentes fases del proceso. A través de un modelo de verificación jerárquico, cada capa de validación aporta un nivel adicional de confianza y transparencia, lo que es especialmente útil en sectores como el inmobiliario, el urbano y el de datos digitales.

En su propuesta, *blockchain* actúa como un registro inmutable, donde cada usuario almacena y mantiene copias de la información, eliminando la necesidad de una autoridad central. Esta estructura descentralizada es aplicada a mercados de datos, asegurando que los datos almacenados y compartidos sean auténticos, verificables y trazables.

La combinación de *waterfall* con *blockchain* representa un enfoque innovador para mejorar la transparencia, seguridad y trazabilidad en proyectos. Sin embargo, su implementación enfrenta desafíos, principalmente por la rigidez del modelo, lo cual dificulta la integración con tecnologías descentralizadas y dinámicas como *blockchain*. Además, los costes de implementación y la falta de regulación específica limitan su adopción a gran escala. A pesar de estos retos, su potencial es significativo si se desarrollan estrategias híbridas que combinen la estructura de *waterfall* con mayor flexibilidad, permitiendo una transición progresiva hacia sistemas más eficientes y confiables.

1.3.2. **PMBOK (PMI)**

El PMBOK (*Project Management Body of Knowledge*) es un conjunto de estándares, mejores prácticas, herramientas y procesos desarrollados por el *Project Management Institute* (PMI) para la

gestión de proyectos [5]. Su propósito es proporcionar un marco metodológico que permita planificar, ejecutar y cerrar proyectos de manera eficiente y efectiva.
El PMBOK (según su sexta edición) está estructurado en diez áreas de conocimiento [53], que cubren los aspectos clave de la gestión de proyectos:

- Gestión de la Integración – Coordina todos los elementos del proyecto.
- Gestión del Alcance – Define qué se incluye y qué no en el proyecto.
- Gestión del Tiempo – Planifica y controla los cronogramas del proyecto.
- Gestión de los Costos – Estima, planifica y controla el presupuesto.
- Gestión de la Calidad – Asegura que los entregables cumplan con los estándares requeridos.
- Gestión de los Recursos – Asigna y administra los recursos humanos y materiales.
- Gestión de la Comunicación – Maneja la distribución y flujo de información del proyecto.
- Gestión de los Riesgos – Identifica, analiza y responde a riesgos del proyecto.
- Gestión de las Adquisiciones – Administra contratos, proveedores y compras.
- Gestión de los Interesados – Maneja la relación con todas las partes involucradas.

Andre Gholam [54] menciona varias aplicaciones de *blockchain* en la gestión de proyectos, incluyendo:

- Automatización de procesos con contratos inteligentes.
- Registro inmutable de información del proyecto para trazabilidad y transparencia.
- Optimización de pagos y adquisiciones a través de *blockchain*.
- Gestión de contratos mediante contratos inteligentes.
- Reducción de disputas gracias a registros inalterables.
- Aplicaciones en la construcción, cadena de suministro y gestión financiera.

También se menciona que el *Blockchain Research Institute* [55] considera que esta tecnología podría transformar las oficinas de gestión de proyectos (PMOs, por sus siglas en inglés) al automatizar la gestión de registros y reportes.

En base a lo mencionado, se propone que *blockchain* puede mejorar la aplicación de PMBOK al proporcionar transparencia, seguridad y automatización en la gestión de proyectos. Los contratos inteligentes, la trazabilidad de datos y la eliminación de intermediarios son algunas de las herramientas que pueden optimizar procesos en áreas clave como gestión de costos, adquisiciones, comunicación y riesgos. Además, la implementación de *blockchain* en proyectos bajo PMBOK permitiría una ejecución más eficiente, con menos errores y mayor control sobre los procesos, lo que lo convierte en una tecnología prometedora para el futuro de la gestión de proyectos.

1.3.3. **ICB (IPMA)**

La IPMA *Individual Competence Baseline* (ICB) [56] es un estándar de competencias en gestión de proyectos que ha sido desarrollado y refinado por la IPMA [7] a lo largo de los años. La versión más reciente, ICB4, proporciona una estructura integral que ayuda a los profesionales de la gestión de

proyectos a desarrollar las habilidades necesarias para liderar proyectos con éxito en diferentes industrias y contextos.

El marco ICB4 se basa en tres áreas clave de competencias [56]:

- **Perspectiva**: se enfoca en el entorno organizacional y estratégico en el que se desarrollan los proyectos. Incluye aspectos como la alineación con la estrategia empresarial, la gobernanza de proyectos y la influencia del entorno externo.

- **Personas**: describe las competencias interpersonales necesarias para la gestión de proyectos, incluyendo liderazgo, comunicación, negociación y toma de decisiones.

- **Práctica**: comprende las herramientas, metodologías y técnicas que permiten ejecutar proyectos de manera estructurada y eficiente. Esto incluye la gestión de riesgos, planificación, control de costos y calidad, entre otros.

ICB no prescribe herramientas o metodologías específicas, sino que establece un marco de referencia adaptable a diferentes enfoques de gestión de proyectos, incluyendo enfoques ágiles, híbridos y tradicionales.

A pesar del creciente interés en la aplicación de *blockchain* en la gestión de proyectos, hasta el momento no se ha documentado una integración formal de *blockchain* dentro de esta metodología.

1.3.4. **PRINCE2**

PRINCE2 (Projects IN Controlled Environments) es una metodología estructurada para la gestión de proyectos, desarrollada originalmente por la Oficina de Comercio Gubernamental del Reino Unido (OGC) y actualmente gestionada por AXELOS. Esta metodología se caracteriza por proporcionar un marco integral basado en principios, procesos y temas claramente definidos, lo que facilita una gestión sistemática y controlada del proyecto [8].

La metodología PRINCE2 se fundamenta en siete principios esenciales que guían cada etapa y decisión durante el ciclo de vida del proyecto [8]:

- **Justificación continua del negocio**: todos los proyectos deben mantener una razón válida durante toda su duración.

- **Aprender de la experiencia**: las lecciones aprendidas se documentan y aplican continuamente para mejorar la eficacia.

- **Roles y responsabilidades definidos**: cada miembro del proyecto tiene responsabilidades claramente establecidas.

- **Gestión por fases**: el proyecto se divide en etapas controladas, cada una con objetivos específicos y puntos de revisión.

- **Gestión por excepción**: se delega autoridad, interviniendo únicamente cuando se superan límites previamente definidos.

- **Enfoque en productos**: se centra en entregar productos concretos y definidos claramente.

- **Adaptación al contexto del proyecto**: permite ajustes según las circunstancias específicas del proyecto.

La gestión mediante PRINCE2 se implementa a través de siete procesos clave que abarcan desde la preparación inicial hasta el cierre formal del proyecto: inicio del proyecto, dirección del proyecto, inicio de una fase, control de una fase, gestión de la entrega de productos, gestión de límites de fase y cierre del proyecto.

En los últimos años, ha crecido el interés por explorar cómo la tecnología *blockchain* puede complementar metodologías estructuradas como PRINCE2. *Blockchain* ofrece potenciales mejoras en transparencia, trazabilidad y automatización, que pueden fortalecer significativamente ciertos principios y procesos clave de PRINCE2 [57].

En la revisión sistemática realizada por Perera et al. [58] se destaca que *blockchain* puede mejorar significativamente la gestión por fases al proporcionar un registro inmutable y verificable de los hitos alcanzados, reduciendo la necesidad de auditorías manuales y aumentando la transparencia entre todas las partes interesadas. De manera similar, Dong et al. [59] destacan que la implementación de contratos inteligentes permite automatizar la gestión por excepción, enviando automáticamente alertas y ejecutando acciones predefinidas cuando se superan los límites establecidos, lo que optimiza la eficiencia operativa y reduce los tiempos de respuesta en la toma de decisiones críticas.

Además, el principio de aprendizaje continuo podría beneficiarse considerablemente de *blockchain* mediante un registro descentralizado y accesible que documente sistemáticamente todas las experiencias previas, facilitando la extracción y reutilización eficiente de conocimientos y lecciones aprendidas en futuros proyectos [57].

Sin embargo, a pesar de los beneficios identificados, la implementación práctica de *blockchain* en entornos gestionados con PRINCE2 sigue enfrentando desafíos importantes. La complejidad tecnológica, los costos iniciales de adopción y la necesidad de adaptación cultural y organizacional son barreras significativas a considerar. Por tanto, aunque la literatura académica reconoce el alto potencial de *blockchain* para complementar PRINCE2, aún se requieren más investigaciones y casos prácticos para validar su aplicación efectiva y comprender mejor cómo superar estas barreras [58].

1.3.5. **Lean**

La metodología *Lean* [60,61] es un enfoque de gestión basado en la eliminación sistemática de desperdicios y la mejora continua de procesos para maximizar el valor entregado al cliente con el mínimo uso de recursos. Se originó en el sistema de producción de Toyota [62] y se ha extendido a múltiples sectores, incluyendo manufactura, desarrollo de software y gestión de proyectos. Lean enfatiza la eficiencia operativa, la optimización del flujo de trabajo y la adaptación flexible a las necesidades del mercado mediante principios como la mejora continua (*Kaizen*), la producción ajustada (*Just in Time*) y el respeto por las personas.

En la publicación de Najafi et al. [63], se explora el potencial de la implementación de *blockchain* en Lean para la mejora de la gestión de datos y la transparencia en procesos de calidad. Los autores destacan que la tecnología *blockchain* puede contribuir a metodologías de mejora continua de la siguiente manera [63]:

- **Trazabilidad y transparencia**: la descentralización de *blockchain* permite registrar cada transacción o modificación en un proceso productivo de manera inmutable, lo que facilita la transparencia y la reducción de errores en la cadena de valor de Lean.

- **Gestión segura de datos**: dado que *Lean* se enfoca en la eliminación de desperdicios y la mejora de la eficiencia, el uso de *blockchain* puede garantizar la seguridad y la integridad de los datos utilizados en la toma de decisiones basada en evidencia.

- **Automatización de procesos con *smart contracts***: la combinación de *blockchain* con *smart contracts* puede permitir la ejecución automática de acciones dentro de los flujos de trabajo de Lean, eliminando tareas innecesarias y asegurando la implementación de mejoras en tiempo real.

- **Integración con *Six Sigma***: en la revisión de casos de estudio, los autores observaron que los modelos de *Six Sigma*, podrían beneficiarse de la implementación de *blockchain* en la gestión de datos, la monitorización de indicadores clave de rendimiento (*KPIs*) y la validación de mejoras de manera descentralizada.

En otra publicación, realizada por Florian et al. [64], los autores del documento proponen una arquitectura Lean para la ejecución descentralizada de procesos mediante *blockchain*. Su enfoque se basa en minimizar la complejidad y los costos operativos, asegurando que solo las partes críticas del proceso sean ejecutadas dentro de la cadena de bloques. Las principales conclusiones de su trabajo fueron:

- **Reducción de costes**: la ejecución parcial dentro y fuera de la *blockchain* disminuye significativamente los costos de transacción, lo que hace viable su adopción en entornos empresariales Lean.

- **Mejora en la trazabilidad y seguridad**: la implementación de contratos inteligentes en *blockchain* garantiza una auditoría segura y automática de los procesos, eliminando la necesidad de intermediarios.

- **Balance entre descentralización y eficiencia**: el uso de *blockchain* de consorcio ofrece un punto medio entre control y descentralización, permitiendo la colaboración entre organizaciones sin sacrificar rendimiento.

- **Aplicabilidad en entornos empresariales**: la propuesta es viable para empresas que buscan implementar *blockchain* sin comprometer la agilidad y eficiencia de sus procesos Lean.

Por otra parte, Perboli et al. [65] concluyeron que la implementación de *blockchain* en la cadena de suministro bajo un enfoque Lean permite no solo mejorar la eficiencia operativa y reducir costes, sino también incrementar la transparencia y confiabilidad del sistema. Sin embargo, destaca que el éxito de la iniciativa depende de la participación de todos los actores involucrados y de una estrategia clara para la adopción de la tecnología. Finalmente, se subraya la importancia de seleccionar cuidadosamente qué información debe registrarse en la *blockchain* para garantizar un equilibrio entre transparencia, seguridad y costes operativos.

1.3.6. **Metodologías Ágiles (*Agile*)**

Las metodologías ágiles son enfoques de gestión de proyectos que enfatizan la flexibilidad, la colaboración y la capacidad de adaptación ante cambios durante el proceso de desarrollo. Surgieron como una alternativa a las metodologías tradicionales, que suelen ser más rígidas y secuenciales. En 2001, se formalizaron los principios fundamentales de estas metodologías en el "Manifiesto por el desarrollo ágil de software" [66], el cual prioriza:

- Individuos e interacciones sobre procesos y herramientas.
- Software funcionando sobre documentación extensiva.
- Colaboración con el cliente sobre negociación contractual.
- Respuesta ante el cambio sobre seguir un plan.

Estas metodologías promueven el desarrollo iterativo e incremental, entregas frecuentes de software funcional y una comunicación constante con el cliente, permitiendo adaptarse rápidamente a los cambios en los requerimientos y mejorando la calidad del producto final.

Entre las metodologías ágiles más destacadas se encuentran Scrum, que se centra en la gestión de proyectos mediante ciclos cortos llamados "sprints", y *Extreme Programming* (XP), que enfatiza prácticas técnicas como la programación en pareja y la integración continua [9].

La implementación de metodologías ágiles ha demostrado ser beneficiosa en diversos contextos, ya que facilita una gestión flexible ante los cambios, fomenta el aprendizaje en equipo y promueve una mayor implicación del cliente en el proceso de desarrollo.

En cuanto a la integración de *blockchain* en estas metodologías, Udvaros et al. [67] sugieren la integración de *blockchain* y contratos inteligentes en metodologías ágiles, específicamente en Scrum, para mejorar la transparencia, seguridad y confianza entre los diferentes actores involucrados en el desarrollo de software. Se sugiere utilizar contratos inteligentes como una forma de automatizar tareas clave dentro del proceso ágil, como la verificación de criterios de aceptación y la ejecución de pagos. La implementación se basa en el uso de *blockchain* como un registro inmutable en el que se almacenan los acuerdos entre clientes, *product owners* y desarrolladores. Cada historia de usuario se vincula a un contrato inteligente que contiene los requisitos y criterios de aceptación. Cuando el desarrollo es completado y validado mediante pruebas criptográficas, el contrato inteligente libera automáticamente los pagos predefinidos.

Las principales ventajas de esta propuesta incluyen [67]:

- Automatización y eliminación de intermediarios, reduciendo errores y tiempos de espera en aprobaciones y pagos.

- Mayor confianza entre las partes, ya que los pagos solo se ejecutan si los criterios acordados son cumplidos.

- Seguridad e inmutabilidad, evitando fraudes o alteraciones en los acuerdos.

- Transparencia, ya que todas las transacciones y validaciones quedan registradas en la *blockchain* y pueden ser auditadas.

Este modelo permite integrar contratos inteligentes sin alterar los procesos ágiles existentes, reforzando los principios de Scrum y asegurando un flujo de trabajo más eficiente y confiable.

En cuanto a su integración con *blockchain*, Farooq et al. [68], proponen AgilePlus, un marco de trabajo basado en *blockchain* para mejorar el desarrollo ágil distribuido (DASD). Su objetivo es resolver problemas de transparencia, confianza, trazabilidad, seguridad y coordinación en equipos de desarrollo de software ubicados en diferentes partes del mundo. Para ello, AgilePlus utiliza una *blockchain* privada de Ethereum, donde los contratos inteligentes (*smart contract*s) automatizan tareas como la aceptación de requisitos, la verificación del trabajo completado y la distribución de pagos a los desarrolladores.

El marco se divide en seis capas: acuerdo, elicitación de requisitos, priorización, diseño y desarrollo, pruebas y pago. En cada fase, los datos y transacciones quedan almacenados en la *blockchain*, garantizando un registro inmutable del progreso del proyecto. Para mejorar la escalabilidad, el sistema usa *Interplanetary File System* (IPFS) como almacenamiento externo, reduciendo la carga de datos en la cadena.

Las principales ventajas de esta integración son: automatización de pagos a través de contratos inteligentes, lo que elimina retrasos y disputas; mayor confianza entre clientes y desarrolladores al registrar todas las interacciones en la *blockchain*; seguridad mejorada gracias a la descentralización y protección contra ataques del 51%; y transparencia total del desarrollo, ya que todas las decisiones y cambios quedan visibles para los participantes [68].

Lenarduzzi et al. [69] exponen un modelo en el que el desarrollo del proyecto se organiza en historias de usuario (*User Stories*, US), las cuales pueden implementarse de manera iterativa en *Sprints* (*Scrum*) o mediante un flujo continuo (*Lean-Kanban*). Cada historia de usuario está vinculada a una prueba de aceptación, cuyo resultado esperado se almacena en un archivo JSON. El *hash* de este archivo es registrado en la *blockchain* utilizando las credenciales criptográficas del *Product Owner* (PO), lo que garantiza transparencia, inmutabilidad y trazabilidad en el proceso de validación.

A diferencia de los enfoques tradicionales, donde la validación de requisitos y pruebas de aceptación depende de la supervisión manual del *Product Owner* y del equipo de calidad, la *blockchain* proporciona una fuente de verdad descentralizada y auditable. Esto elimina posibles alteraciones o manipulaciones, minimiza errores humanos y evita disputas entre el cliente y el equipo de desarrollo sobre el cumplimiento de los requisitos. Además, al ser públicamente accesible y verificable, se reduce la necesidad de intermediarios y se mejora la confianza en el proceso.

Cuando un desarrollador completa una historia de usuario, ejecuta la prueba de aceptación correspondiente. Si el resultado obtenido genera un *hash* idéntico al registrado en la *blockchain*, el *Smart Contract* valida automáticamente la historia de usuario y autoriza el pago al desarrollador en criptomonedas o tokens, asegurando un proceso de retribución justo y automatizado basado en el cumplimiento real de las tareas [69].

1.3.7. **PM²**

La metodología PM² [11] fue desarrollada por la Comisión Europea para proporcionar una estructura eficiente de gestión de proyectos dentro de instituciones gubernamentales, organizaciones del sector privado y equipos de trabajo en la UE. Su propósito principal es estandarizar procesos y garantizar la alineación de proyectos con las estrategias y objetivos organizacionales.

Una característica distintiva de PM² es su accesibilidad, ya que está diseñada para ser utilizada por cualquier organización, no solo por aquellas dentro de la UE. Su flexibilidad la hace aplicable a diversos contextos, desde proyectos tecnológicos hasta iniciativas de transformación digital.
PM² se basa en cuatro dimensiones fundamentales [11]:

- Gobernanza del proyecto: define los roles y responsabilidades dentro del equipo de gestión del proyecto.
- Ciclo de vida del proyecto: establece las fases del proyecto: inicio, planificación, ejecución, monitoreo y cierre.
- Procesos de gestión: incluye procesos esenciales para la gestión de proyectos como planificación, gestión de riesgos, gestión de calidad, etc.
- Artefactos: proporciona plantillas y herramientas para facilitar la documentación y la toma de decisiones.

Una de las ventajas de PM² es que incorpora enfoques ágiles y tradicionales, permitiendo adaptaciones según la naturaleza del proyecto.

Hasta la fecha, no se han identificado estudios o publicaciones que aborden específicamente la implementación de *blockchain* dentro de esta metodología.

Dado el potencial de *blockchain* para mejorar la transparencia, seguridad y automatización de procesos dentro de la gestión de proyectos, sería relevante desarrollar estudios que exploren cómo esta tecnología puede integrarse con los principios y procesos de PM².

1.3.8. **Conclusión Comparativa**

Tras la revisión y análisis de los distintos estudios realizados por otros autores, se concluye que la integración de *blockchain* en la gestión de proyectos ha sido explorada en diversas metodologías, como *Waterfall*, PMBOK (PMI), Lean y Agile, donde se han identificado beneficios claros en términos de trazabilidad, transparencia, automatización y seguridad. Sin embargo, en el caso de las metodologías ICB (IPMA) y PM², no se han encontrado investigaciones que aborden su implementación con *blockchain* de forma específica, lo que representa una evidente brecha en la literatura académica y, al mismo tiempo, una oportunidad de innovación e investigación aplicada.

En este contexto, la metodología ICB (IPMA) no ha sido seleccionada debido a su orientación centrada en competencias individuales del profesional de proyectos. Este enfoque, más personal que estructural, se aleja del propósito de este estudio, que busca integrar la lógica operativa y documental de una metodología con tecnologías distribuidas que requieren procesos claramente definidos y automatizables.

Por el contrario, la metodología PM² ha sido seleccionada por varias razones estratégicas. En primer lugar, PM² ofrece un marco estructurado, abierto y ampliamente documentado, con artefactos concretos, roles bien definidos y un ciclo de vida claro, lo que facilita su descomposición en lógica formal susceptible de ser programada en contratos inteligentes. Además, su diseño modular y adaptable, junto con su versión libre Open PM², la convierte en una opción viable tanto para el sector público como para organizaciones privadas de distintos tamaños y niveles de madurez metodológica.

Asimismo, PM² presenta una clara alineación con los principios que *blockchain* potencia, como la transparencia, la rendición de cuentas, la validación distribuida de procesos y la trazabilidad de decisiones. Su enfoque centrado en resultados, gobernanza efectiva y mejora continua resulta altamente compatible con los beneficios que ofrece la tecnología *blockchain*, especialmente en términos de registro inmutable, automatización de flujos de trabajo mediante *smart contracts* y auditoría descentralizada de acciones clave dentro del proyecto.

Por todo lo anterior, este trabajo de fin de máster se centrará en el diseño de un modelo de integración de *blockchain* en la metodología PM², con el objetivo de analizar cómo esta tecnología puede mejorar la eficiencia, fiabilidad y seguridad en la gestión de proyectos. Para ello, se abordarán casos de uso aplicables, posibles aplicaciones técnicas y los desafíos que plantea su implementación, proponiendo una solución estructurada que optimice los procesos dentro del marco de PM².

Alineado con los objetivos de este trabajo, el estudio busca contribuir al avance del conocimiento en dirección de proyectos mediante una propuesta innovadora, realista y alineada con las tendencias tecnológicas actuales, sentando también las bases para futuras investigaciones académicas y aplicaciones prácticas en el ámbito profesional, tanto en entornos institucionales como corporativos.

Capítulo 2 Metodología

En este capítulo se presenta una descripción detallada de la metodología PM² (*Project Management Methodology*), destacando su relevancia para la integración efectiva de tecnologías emergentes, específicamente *blockchain* y contratos inteligentes. Inicialmente, se expone un panorama general del marco metodológico PM², enfatizando sus principios fundamentales, fases del ciclo de vida del proyecto, roles específicos asignados a cada participante y procesos documentales característicos.

Posteriormente, se profundiza en la selección tecnológica adecuada, argumentando la elección estratégica de Ethereum como red *blockchain* y Solidity como lenguaje de programación debido a su madurez, interoperabilidad y funcionalidad técnica para la implementación práctica. Finalmente, se aborda un análisis sistemático de los artefactos PM² para identificar cuáles resultan más idóneos para ser implementados mediante contratos inteligentes, estableciendo criterios específicos como viabilidad técnica, trazabilidad e impacto en automatización. Esta metodología estructurada asegura una integración coherente y efectiva de *blockchain* con PM², alineándose así con los objetivos estratégicos del presente estudio.

2.1 Metodología PM²

Dado el interés de este trabajo en explorar la integración de *blockchain* en metodologías de gestión de proyectos, y ante la escasa literatura existente sobre su aplicación en PM², se justifica la necesidad de analizar esta metodología en profundidad. A continuación, se presenta una visión general de PM², con el fin de establecer las bases necesarias para su posterior conexión con tecnologías emergentes como *blockchain* y contratos inteligentes.

2.1.1. Introducción a la Metodología PM²

La metodología PM² (*Project Management Methodology*) es un marco metodológico creado por la Comisión Europea [11] con el objetivo de proporcionar un enfoque común y estructurado para la gestión eficiente de proyectos en instituciones públicas, aunque también aplicable en organizaciones privadas y no gubernamentales. Su desarrollo se origina por la necesidad de contar con una metodología clara, simple y robusta que pueda ser aplicada universalmente dentro de los Estados miembros de la Unión Europea, asegurando la transparencia, coordinación y cumplimiento efectivo de los objetivos planteados.

PM² combina aspectos destacados de otras metodologías ampliamente reconocidas (como PMBOK, PRINCE2, IPMA y enfoques ágiles), sintetizándolos en un modelo de gestión accesible, orientado a resultados, adaptable y claramente definido. Además, en su versión abierta, conocida como Open PM², permite que organizaciones de cualquier sector o tamaño puedan adaptar libremente sus principios y procesos para gestionar sus proyectos de manera eficiente y transparente.

2.1.2. Principios Fundamentales

La metodología PM² se basa en una serie de principios clave que guían la ejecución efectiva del proyecto [11]:

- **Orientación a resultados:** el éxito de los proyectos se evalúa en función del cumplimiento efectivo de los resultados y beneficios definidos inicialmente.

- **Transparencia y gobernanza efectiva:** PM² promueve la comunicación abierta y constante, así como mecanismos claros de gobernanza que permiten la supervisión eficaz y la rendición de cuentas.

- **Claridad de roles:** cada integrante del proyecto posee funciones y responsabilidades claramente definidas, lo que facilita la colaboración efectiva y evita conflictos operativos.

- **Participación y colaboración activa:** se enfatiza la importancia del compromiso activo y permanente de todas las partes interesadas involucradas en el proyecto.

- **Mejora continua:** se fomenta la reflexión sobre las experiencias adquiridas durante el desarrollo del proyecto para incorporar lecciones aprendidas en futuros proyectos.

2.1.3. **Ciclo de Vida del Proyecto**

La metodología PM² define un ciclo de vida claro y ordenado compuesto por cuatro fases principales que permiten gestionar eficientemente los proyectos desde su concepción hasta su cierre definitivo [11]:

- **Iniciación:** la fase inicial se centra en establecer la necesidad, alcance preliminar y la viabilidad general del proyecto. Durante esta fase, se identifican las expectativas y requerimientos principales de las partes interesadas, así como los recursos potencialmente necesarios.

- **Planificación:** en esta fase se desarrollan en detalle las actividades y recursos necesarios para ejecutar el proyecto. Se identifican riesgos, restricciones, tiempos y costos previstos, dando lugar a una planificación sólida que sirva de guía para las siguientes fases.

- **Ejecución:** corresponde a la implementación efectiva del plan del proyecto. En esta fase, el equipo de gestión coordina y supervisa las tareas definidas en la planificación, asegurando que el proyecto avance conforme a lo establecido. Asimismo, se gestionan los riesgos e incidencias que surjan en el proceso.

- **Cierre:** durante esta última fase, se realiza la evaluación final del proyecto, verificando el cumplimiento de objetivos, documentando los resultados obtenidos, y recopilando experiencias para futuras iniciativas. Posteriormente, se realiza la entrega formal del producto o servicio resultante al área o cliente final correspondiente.

Cada fase está respaldada por una serie de procesos de gestión (por ejemplo, gestión del alcance, de riesgos, del tiempo, del cambio o de interesados) y por un conjunto de artefactos (documentos estructurados, plantillas o registros) que recogen la información relevante, aseguran la trazabilidad y estandarizan la ejecución. Este enfoque basado en procesos y artefactos distingue a PM² como una metodología rigurosa, pero al mismo tiempo adaptativa, permitiendo su uso en contextos muy diversos.

2.1.4. **Roles y Responsabilidades**

Una de las características distintivas de PM² es su enfoque claramente estructurado (ver Figura 6) respecto a la asignación de roles y responsabilidades, asegurando que todos los participantes entiendan perfectamente sus funciones dentro del proyecto [11]:

- ***Project Owner* (PO):** máximo responsable del proyecto desde una perspectiva estratégica, encargado de la toma de decisiones fundamentales y aprobación del resultado final.
- ***Project Steering Committee* (PSC):** comité directivo que supervisa la evolución del proyecto y apoya la toma de decisiones relevantes para asegurar la alineación estratégica
- ***Business Manager* (BM):** responsable del cumplimiento de objetivos estratégicos, garantizando que el proyecto genere valor organizacional y cumpla con los resultados esperados.
- ***Project Manager* (PM):** encargado directo de gestionar las actividades diarias del proyecto, administrar recursos, gestionar riesgos y resolver incidencias operativas.
- ***Solution Provider Team* (SPT):** grupo especializado en proporcionar las soluciones técnicas o funcionales necesarias para el éxito del proyecto.
- ***Project Core Team* (PCT):** personal operativo asignado directamente a la realización efectiva de las tareas del proyecto bajo la coordinación del Project Manager.
- ***Stakeholders*:** partes interesadas internas o externas cuyo interés o influencia pueden afectar o verse afectados significativamente por los resultados del proyecto.

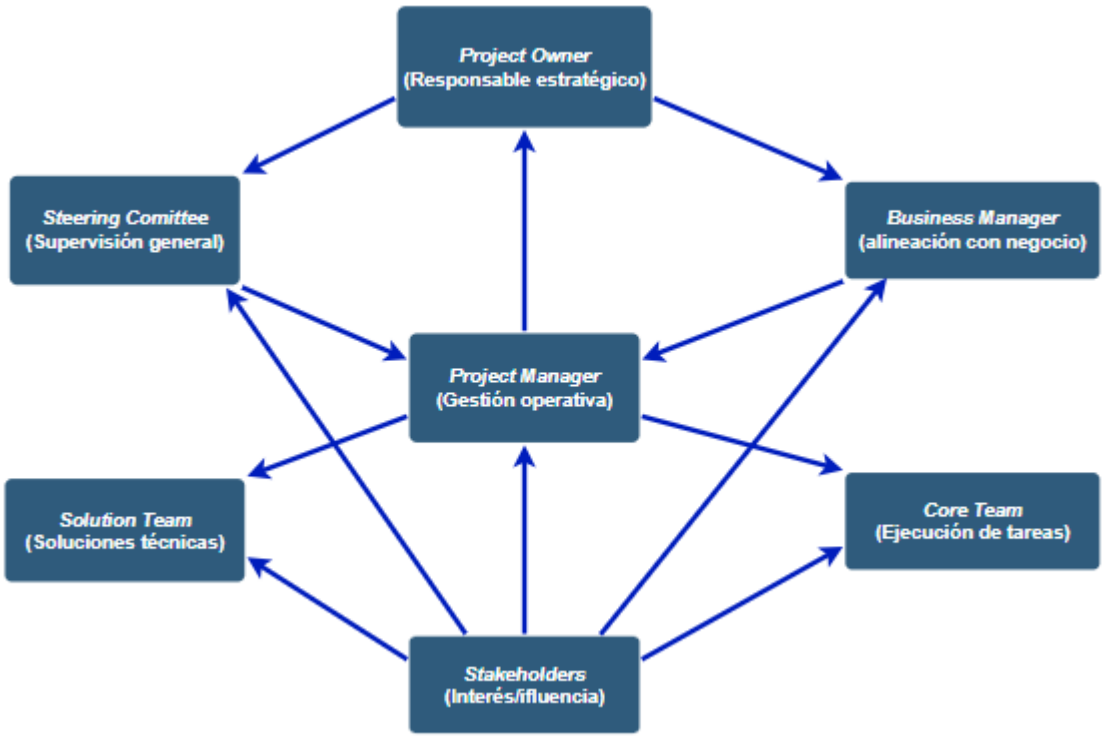

Figura 6. Estructura de roles y responsabilidades en PM^2. Elaboración propia en diagrams.net [70].

Esta estructura organizativa claramente definida garantiza una comunicación fluida, decisiones informadas y una ejecución eficiente de tareas en cada fase.

2.1.5. **Procesos de Gestión y Documentación en PM²**

La metodología PM² se fundamenta en un enfoque estructurado de gestión por procesos y artefactos documentales. Este modelo permite una planificación clara, una ejecución controlada y una trazabilidad efectiva de las actividades y decisiones del proyecto a lo largo de su ciclo de vida.

PM² define una serie de procesos de gestión, alineados con las buenas prácticas en dirección de proyectos, que abarcan áreas como la gestión del alcance, el tiempo, los costes, la calidad, los riesgos, los interesados, la comunicación y el cambio. Estos procesos están presentes a lo largo de las cuatro fases del ciclo de vida del proyecto: iniciación, planificación, ejecución y cierre.

Para apoyar estos procesos, PM² proporciona una colección estandarizada de 33 artefactos [11], que constituyen el sistema documental de la metodología. Los artefactos son documentos, plantillas y registros que capturan, comunican y controlan la información del proyecto. Están clasificados en tres grandes categorías según su función:

- 15 artefactos de gestión que dan soporte directo a la planificación, supervisión y control del proyecto.
- 6 artefactos relacionados con los entregables, los cuales son documentos que apoyan a la producción, validación y transferencia de los entregables.
- 12 artefactos de soporte que actúan como documentación adicional que apoya procesos específicos y mejora la comunicación y gestión de partes interesadas.

La Figura 7 muestra el desglose de la clasificación recién descrita.

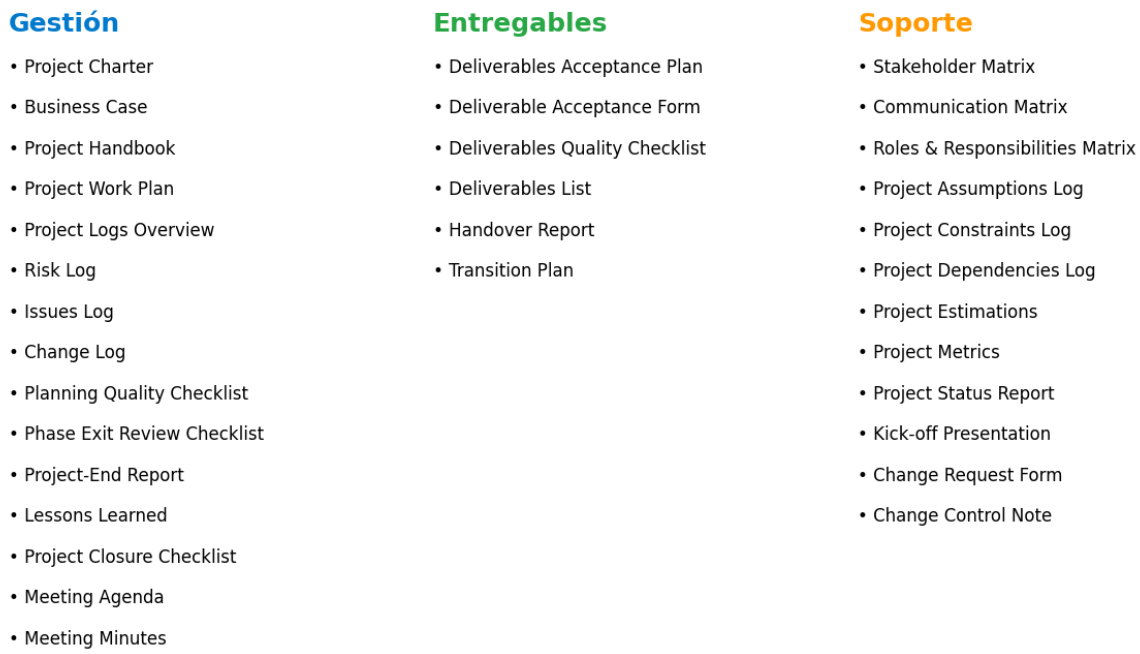

Figura 7. Clasificación de artefactos en la metodología PM². Elaboración propia con Python [36].

Cada uno de estos artefactos cumple una función específica dentro del sistema de gestión del proyecto. Se utilizan en distintos momentos del ciclo de vida del proyecto, y su implementación contribuye a

estandarizar la información, garantizar la trazabilidad de decisiones, facilitar auditorías internas y externas y mejorar la comunicación entre roles y partes interesadas.

Por lo que el uso coherente de estos documentos proporciona una base sólida para la gobernanza del proyecto, permitiendo una gestión más profesional, transparente y alineada con los objetivos estratégicos de la organización. Además, PM² ofrece plantillas oficiales descargables desde el portal de Open PM², lo que facilita su adopción incluso en entornos con poca experiencia previa en metodologías formales.

2.1.6. **Herramientas y Técnicas en PM²**

Complementando su marco documental y su enfoque por procesos, PM² incorpora un conjunto de herramientas y técnicas de gestión que permiten estructurar las actividades del proyecto, mitigar riesgos y facilitar una toma de decisiones informada. Estas técnicas se aplican de forma transversal y pueden adaptarse a proyectos de distinta naturaleza y complejidad.

Entre las herramientas más destacadas se encuentran [11]:

- **Descomposición del trabajo (WBS – *Work Breakdown Structure*)**: técnica fundamental para dividir el alcance del proyecto en componentes más manejables, facilitando la asignación de tareas, recursos y responsabilidades.

- **Cronogramas y diagramas de Gantt**: herramientas visuales para planificar el tiempo, establecer dependencias entre actividades y controlar el progreso.

- **Gestión de riesgos**: PM² recomienda identificar riesgos de forma temprana, analizar su impacto y probabilidad, y planificar medidas preventivas y correctivas. Esta gestión se apoya en artefactos como el *Risk Log* y matrices de riesgo.

- **Matrices de interesados (*Stakeholder Matrix*)**: técnica para mapear la influencia e interés de las partes interesadas, diseñando estrategias de comunicación adaptadas a cada grupo.

- **Gestión del cambio**: PM² establece un proceso estructurado que incluye herramientas como el *Change Request Form* y el *Change Log* para documentar, evaluar y aprobar cambios en alcance, cronograma, presupuesto o calidad.

- **Técnicas ágiles integradas**: aunque PM² no es una metodología ágil, permite la adopción de prácticas ágiles como *daily stand-ups*, retrospectivas, tableros Kanban o entregas incrementales, especialmente útiles en proyectos tecnológicos o de innovación con alta incertidumbre.

Estas herramientas, junto con el uso estructurado de los artefactos y la aplicación disciplinada de los procesos, conforman un marco metodológico integral y escalable, que permite una gestión eficiente y profesional de proyectos, al tiempo que facilita la integración de tecnologías emergentes como *blockchain* y contratos inteligentes.

2.1.7. **Adaptabilidad del Modelo PM²**

Una característica destacable de PM² es su alta adaptabilidad, permitiendo una fácil personalización según las necesidades específicas de cada proyecto y organización. Esta flexibilidad se manifiesta en:

- **Escalabilidad del modelo**: PM² puede adaptarse tanto a proyectos de pequeña escala como a iniciativas complejas con múltiples actores involucrados.
- **Integración con otras prácticas**: la metodología admite la incorporación de elementos procedentes de enfoques ágiles, predictivos o híbridos, según el contexto particular del proyecto.
- **Aplicabilidad sectorial diversa**: aunque inicialmente diseñada para instituciones públicas europeas, su estructura modular facilita su adopción en múltiples sectores como tecnología, sanidad, construcción, educación y servicios profesionales.

Gracias a esta flexibilidad, PM² es especialmente apta para integrarse con tecnologías emergentes como *blockchain*, ofreciendo un marco sólido y adaptable para la implementación de soluciones tecnológicas innovadoras.

2.1.8. **Comparativa General con Otras Metodologías**

Antes de avanzar con la aplicación práctica de la metodología PM² en este trabajo, resulta pertinente contextualizar su propuesta mediante una comparación estructurada con otras metodologías ampliamente utilizadas en la dirección de proyectos. Este ejercicio no solo permite posicionar a PM² dentro del panorama metodológico actual, sino también argumentar su idoneidad frente a alternativas tradicionales, como PMBOK, PRINCE2 o el modelo en cascada, así como frente a enfoques más recientes como Lean o las metodologías ágiles.

La Tabla 1 presenta una comparativa basada en cinco criterios fundamentales: estructura organizativa, nivel de documentación requerida, adaptabilidad, complejidad de uso y aplicabilidad sectorial. Este análisis permite identificar con mayor precisión las fortalezas de PM², particularmente su capacidad para combinar rigor metodológico con flexibilidad operativa, cualidades esenciales en entornos que requieren la integración de tecnologías emergentes como *blockchain*.

En conjunto, la comparativa destaca a PM² como una metodología equilibrada, capaz de incorporar lo mejor de los enfoques tradicionales y ágiles. Esto la convierte en una opción especialmente adecuada para organizaciones que buscan una estructura sólida sin renunciar a la capacidad de adaptación.

Tabla 1. Comparativa de metodologías de dirección de proyectos según aspectos clave. Elaboración propia con Microsoft Excel [71].

Aspecto	PM^2	PMBOK (PMI)	ICB (IPMA)	PRINCE2	Cascada	Lean	Ágiles (Scrum, Kanban...)
Estructura organizativa	Clara y definida, con roles precisos	Flexible y adaptable a distintas estructuras	Centrada en competencias individuales, no prescriptiva	Estructurada, basada en roles y jerarquías	Jerárquica y secuencial	Horizontal, orientada a flujo y valor	Flexible, con equipos autoorganizados
Documentación requerida	Moderada, con plantillas estándar	Amplia y detallada	Variable, depende del contexto	Muy exhaustiva y formalizada	Elevada, secuencial y previa a ejecución	Mínima, enfocada a valor y mejora continua	Ligera, adaptativa y centrada en el producto
Adaptabilidad	Alta, permite enfoques híbridos	Alta, con guías adaptativas	Muy alta, adaptable a marcos ágiles o tradicionales	Moderada, requiere disciplina estructurada	Baja, rígida por naturaleza secuencial	Alta, adaptable a procesos productivos o de servicios	Muy alta, pensada para entornos cambiantes
Complejidad de uso	Moderada, fácil de implementar	Alta, requiere conocimiento técnico	Alta, centrada en desarrollo de competencias	Alta, por su nivel de formalismo y control	Alta, por su linealidad y escasa flexibilidad	Moderada, requiere compromiso cultural	Baja, fácil de adoptar con equipos formados
Aplicabilidad sectorial	Muy versátil, tanto en sector público como privado	Universal, aplicable en cualquier sector	Flexible, centrado en desarrollo profesional	Predominantemente en sectores regulados	Ingeniería, construcción, desarrollo tradicional	Industria, manufactura, servicios orientados al cliente	Sectores tecnológicos, innovación, entornos inciertos

2.1.9. **Ventajas, Limitaciones y Aplicabilidad de PM²**

Como cualquier metodología, PM² presenta ventajas y también ciertas limitaciones a considerar:

Ventajas principales:

- **Gratuita y abierta**: acceso libre mediante la versión Open PM², lo que facilita su difusión y adopción.
- **Claridad organizativa**: ofrece una estructura sencilla con roles claramente definidos, facilitando su implementación y gobernanza.
- **Orientada a resultados**: foco en el valor y beneficios obtenidos, facilitando la gestión efectiva del proyecto.
- **Alta adaptabilidad**: permite ajustes específicos según las necesidades concretas del proyecto o la organización.

Limitaciones identificadas:

- **Difusión limitada**: relativamente nueva y menos conocida fuera del contexto europeo, lo que dificulta su reconocimiento y adopción internacional.
- **Menor nivel de madurez**: en comparación con metodologías consolidadas como PMBOK o PRINCE2, PM² todavía se encuentra en una etapa temprana de adopción generalizada.
- **Requiere adaptación para contextos ágiles puros**: aunque admite técnicas ágiles, no es una metodología enteramente ágil, por lo que puede necesitar ajustes adicionales en contextos extremadamente dinámicos o complejos.

La elección de PM² como marco metodológico central en este trabajo se sustenta en varios factores estratégicos relacionados directamente con el objetivo general del estudio:

- PM² proporciona una estructura metodológica clara y sólida, ideal para explorar la integración efectiva de tecnologías disruptivas como *blockchain*.
- Su orientación hacia la transparencia, la rendición de cuentas y la gobernanza efectiva, se alinea directamente con las ventajas inherentes a la tecnología *blockchain*.
- Los procesos claramente definidos en PM² permiten explorar cómo los contratos inteligentes pueden optimizar la automatización de la gestión contractual, la validación de entregables, y el control eficiente de hitos y fases en los proyectos.
- La adaptabilidad inherente del modelo PM² facilita que su integración con *blockchain* pueda realizarse de manera escalable, ya sea en proyectos públicos o privados, promoviendo su generalización futura.

De este modo, la relevancia del marco PM² radica en su capacidad de servir como plataforma metodológica sólida, sobre la cual este TFM diseñará una propuesta innovadora para integrar *blockchain* en la gestión de proyectos, contribuyendo así al avance del conocimiento y ofreciendo una solución práctica acorde con las tendencias tecnológicas actuales.

2.2 Elección de Red *Blockchain* y Lenguaje de Programación para la Implementación en Artefactos PM²

La selección de la infraestructura tecnológica adecuada es un aspecto crítico para el éxito de cualquier propuesta basada en *blockchain*, especialmente cuando se trata de automatizar procesos vinculados a artefactos metodológicos como los definidos en PM². En este contexto, la red Ethereum, combinada con el lenguaje de programación Solidity, representa la opción más adecuada en términos de madurez, funcionalidad, ecosistema y viabilidad práctica.

Ethereum [28] es la red *blockchain* pública más consolidada en lo que respecta al desarrollo de contratos inteligentes. Desde su lanzamiento en 2015, ha evolucionado hasta convertirse en el estándar de facto para aplicaciones descentralizadas (dApps), y su infraestructura ha sido adoptada por miles de proyectos en múltiples sectores, incluidos el financiero, logístico, sanitario y gubernamental.

Las principales razones por las que se elige Ethereum son:

1. **Madurez tecnológica y comunidad activa**: Ethereum cuenta con el ecosistema de desarrollo más amplio y robusto del sector *blockchain*. Dispone de una vasta comunidad internacional, herramientas consolidadas, bibliotecas auditadas y entornos de desarrollo integrados como Remix, Truffle o Hardhat, que facilitan la creación, prueba y despliegue de contratos inteligentes incluso en entornos académicos o institucionales.

2. **Soporte completo para lógica contractual compleja**: a través de la *Ethereum Virtual Machine* (EVM), esta red permite la ejecución de contratos inteligentes altamente expresivos, capaces de automatizar flujos de trabajo, validar entregables, registrar eventos, gestionar autorizaciones o ejecutar acciones condicionadas. Estas capacidades son esenciales para representar digitalmente artefactos como el *Change Log*, *Risk Log*, *Deliverable Acceptance Form* o *Project Work Plan*, donde es necesario asegurar trazabilidad, inmutabilidad y ejecución automática de reglas.

3. **Interoperabilidad y compatibilidad futura**: Ethereum ha sido adoptado como base tecnológica por una gran cantidad de redes alternativas que mantienen compatibilidad con la EVM (por ejemplo, Polygon, Binance Smart Chain, Avalanche, entre otras). Esto permite que cualquier solución desarrollada sobre Ethereum pueda migrarse fácilmente a otras redes con costes más bajos, sin necesidad de reescribir el código. Además, la compatibilidad con estándares como ERC-20, ERC-721 o ERC-1155 facilita la integración con otros sistemas, como plataformas de gestión documental, identidad digital o tokenización de entregables.

4. **Accesibilidad y validación pública**: como red pública y sin permisos, Ethereum permite el acceso abierto, la auditoría descentralizada y la validación externa de la información registrada. Esto es especialmente relevante cuando se busca reforzar la transparencia y gobernanza en contextos públicos o colaborativos, donde la confianza entre partes no puede darse por supuesta.

Por la parte del lenguaje de programación, Solidity es el más utilizado para el desarrollo de contratos inteligentes en el entorno Ethereum y redes compatibles con EVM. Diseñado específicamente para este propósito, combina expresividad funcional con facilidad de adopción, siendo actualmente el lenguaje más documentado y con mayor soporte en el ámbito *blockchain*. Los motivos que justifican su elección son los siguientes:

1. **Especialización para contratos inteligentes**: Solidity permite definir estructuras de datos complejas, controlar flujos condicionales, restringir el acceso a funciones, emitir eventos y gestionar de forma segura los estados de un contrato. Estas funcionalidades permiten modelar con precisión los requisitos de un artefacto PM², desde la validación de entregables hasta la autorización de cambios o el registro histórico de decisiones.

2. **Amplia documentación y curva de aprendizaje razonable**: existen múltiples recursos formativos, tutoriales, entornos de pruebas en línea (como Remix IDE) y herramientas de simulación. Esto facilita su aprendizaje y adopción incluso por equipos que no tienen experiencia previa en programación *blockchain*, lo que resulta clave en proyectos piloto, entornos académicos o fases de prototipado.

3. **Ecosistema técnico y bibliotecas auditadas**: Solidity cuenta con un ecosistema maduro que incluye librerías estándar como OpenZeppelin, que ofrecen componentes seguros y reutilizables para funciones críticas como control de acceso, gestión de roles, validación de firmas, y otras prácticas comunes. El uso de estas bibliotecas permite reducir los riesgos de seguridad asociados al desarrollo de contratos.

4. **Integración con interfaces externas**: Solidity permite la emisión de eventos y la exposición de funciones públicas que pueden ser invocadas desde sistemas externos, como plataformas de gestión de proyectos, bases de datos institucionales o paneles de visualización. Esta capacidad es clave para lograr una integración práctica con los sistemas que actualmente gestionan los artefactos de PM² en su formato tradicional.

Durante el análisis se evaluaron otras redes y lenguajes, que fueron descartados por motivos técnicos o estratégicos:

- **Hyperledger Fabric (Go/Java)**: excelente para entornos empresariales privados, pero requiere una infraestructura compleja con nodos autorizados, lo que dificulta su implementación en entornos públicos, colaborativos o académicos. Además, no proporciona interoperabilidad ni validación pública.

- **Solana (Rust)**: destaca por su rendimiento, pero sufre de menor estabilidad, una curva de aprendizaje elevada debido a su lenguaje de programación, y no es compatible con EVM, lo que limita su integración con herramientas estándar.

- **Sui / Aptos (Move)**: aunque prometedoras en cuanto a escalabilidad y seguridad formal, su juventud, falta de documentación consolidada y baja adopción las convierten en opciones poco viables para desarrollos orientados a la producción o validación institucional.

- **Polkadot / Cosmos (Rust / CosmWasm)**: enfocadas en interoperabilidad multichain, pero requieren altos conocimientos técnicos y complejos modelos de despliegue. Además, no disponen de un entorno de contratos inteligentes tan maduro ni accesible como Ethereum.

La elección de Ethereum como red y Solidity como lenguaje de programación se justifica por su equilibrio entre madurez tecnológica, expresividad funcional, soporte comunitario y adaptabilidad.

Esta combinación proporciona la infraestructura necesaria para automatizar, auditar y asegurar la gestión de artefactos clave de la metodología PM² mediante contratos inteligentes, al tiempo que garantiza la posibilidad de escalar, integrar o migrar la solución en el futuro.

Por todo ello, se trata de la opción más sólida, viable y estratégica para el diseño e implementación del modelo de integración *blockchain* propuesto en este Trabajo de Fin de Máster.

2.3 Análisis de Artefactos PM² para Implementación con Contratos Inteligentes

Como se ha descrito anteriormente, la metodología PM² contempla un conjunto de 33 artefactos, agrupados en tres grandes categorías: artefactos de gestión, artefactos relacionados con los entregables y artefactos de soporte. Cada artefacto desempeña un rol específico dentro del ciclo de vida del proyecto, desde la fase inicial de planificación hasta la fase final de cierre.

Sin embargo, no todos los artefactos poseen la misma idoneidad para ser implementados mediante tecnología *blockchain* utilizando contratos inteligentes. Dada la especificidad técnica de Ethereum y Solidity, se han identificado ciertos criterios clave que permiten evaluar su viabilidad de implementación y el valor añadido de dicha automatización *blockchain*.

Para determinar qué artefactos resultan aptos para su implementación mediante contratos inteligentes, se han definido tres criterios fundamentales, que permiten justificar la clasificación realizada en la tabla posterior:

- **Viabilidad técnica:** se evalúa si la estructura, contenido y lógica del artefacto permiten su representación en código ejecutable dentro de un contrato inteligente. Para ello se analiza:
    - Si el artefacto requiere una interacción lógica automatizable.
    - Si es posible definir reglas de negocio claras y estandarizadas que puedan programarse.
    - Si la información asociada al artefacto se beneficia significativamente de la inmutabilidad *blockchain*.

- **Valor añadido por trazabilidad inmutable:** se examina el beneficio derivado del almacenamiento seguro e inmutable del historial del artefacto en *blockchain*, considerando:
    - La necesidad real de transparencia y auditoría externa.
    - La criticidad del registro inalterable de decisiones o aprobaciones.
    - La mejora en la confianza entre partes al usar un registro descentralizado y verificable.

- **Potencial de automatización lógica:** se evalúa la utilidad real de la automatización mediante lógica *on-chain* (ejecutada directamente sobre la cadena de bloques o *blockchain*), analizando:
    - La posibilidad de automatizar procesos de validación y aprobación.
    - El impacto real en reducción de tareas manuales y administrativas.
    - La capacidad para reducir errores o conflictos mediante ejecución automatizada y condicional de procesos.

Tras aplicar los criterios recién definidos, los 33 artefactos PM² han sido agrupados en cuatro niveles claramente diferenciados:

1. **Muy recomendable**: en este nivel se encuentran aquellos artefactos cuyo valor estratégico y beneficio funcional es considerablemente alto al ser implementados con contratos inteligentes. Su lógica interna permite automatización completa o casi completa, y presentan un gran valor añadido mediante la inmutabilidad *blockchain*.

2. **Sí (recomendable)**: estos artefactos presentan una buena viabilidad técnica, aunque algo menor en términos de automatización total. Su valor principal proviene de la trazabilidad que aporta *blockchain* para auditorías y verificaciones frecuentes.

3. **Parcialmente recomendable**: estos artefactos presentan cierta viabilidad técnica, pero la automatización con *blockchain* debería enfocarse únicamente en determinados aspectos específicos del artefacto. Habitualmente tienen procesos internos parcialmente ejecutables y ofrecen una ventaja moderada en cuanto a trazabilidad o automatización puntual.

4. **No recomendable**: en este grupo se encuentran artefactos cuya naturaleza y uso no ofrecen ventajas significativas ni valor tangible mediante implementación *blockchain*. Son documentos predominantemente descriptivos, informativos o estratégicos sin una lógica clara para automatización. La rigidez técnica del *blockchain* tampoco aporta beneficios específicos a estos artefactos.

Esta clasificación integral asegura una implementación *blockchain* focalizada, estratégica y alineada con los principios fundamentales de eficiencia, seguridad y transparencia que PM² promueve. Al considerar claramente cuatro niveles diferenciados, se garantiza una evaluación precisa y orientada al valor añadido real que ofrece la tecnología *blockchain* en la gestión documental y operativa del proyecto.

La tabla que se muestra a continuación (Tabla 2) resume gráficamente el análisis realizado, proporcionando una clasificación clara y fundamentada para cada artefacto PM², basada en los criterios definidos anteriormente. Este análisis integral facilita la comprensión de por qué ciertos artefactos son ideales para *blockchain* mientras que otros permanecen más eficientemente gestionados mediante métodos tradicionales.

Tabla 2. Análisis de los 33 artefactos de PM² y su viabilidad para ser implementados mediante contratos inteligentes en Ethereum utilizando Solidity. Elaboración propia con Excel [71].

Artefacto	¿Implementabilidad con *Smart Contract*?	Justificación
Project Charter	Muy recomendable	Documento de intención. Requiere flexibilidad y aprobaciones externas.
Business Case	No	Análisis estratégico no automatizable fácilmente.
Project Handbook	No	Guía operativa interna sin lógica ejecutable.
Project Work Plan (PWP)	Parcialmente	Podría registrarse para auditoría, pero no esencialmente automatizable.
Project Logs Overview	Sí	Ideal para consolidar registros inmutables de logs.
Risk Log	Sí	Beneficia de trazabilidad y evaluación inmutable.
Issues Log	Sí	Seguimiento de incidencias con trazabilidad es clave.
Change Log	Muy recomendable	Ideal para gestionar cambios con lógica de aprobación.
Planning Quality Checklist	No	Lista de control interna, sin necesidad de *blockchain*.
Phase Exit Review Checklist	No	Herramienta de revisión sin ejecución lógica.

Artefacto	¿Implementabilidad con *Smart Contract*?	Justificación
Project-End Report	Parcialmente	Puede registrarse como cierre oficial validado.
Lessons Learned	No	Reflexión cualitativa, no útil para *blockchain*.
Project Closure Checklist	No	Control interno sin valor añadido en *blockchain*.
Meeting Agenda	No	No requiere trazabilidad ni lógica ejecutable.
Meeting Minutes	Parcialmente	Si se requieren firmas múltiples, puede ser útil.
Deliverable Acceptance Plan	Muy recomendable	Puede automatizar condiciones de aceptación.
Deliverable Acceptance Form	Muy recomendable	Alta utilidad para formalizar entregables aceptados.
Deliverables Quality Checklist	No	*Checklists* cualitativas no requieren trazabilidad.
Deliverables List	Parcialmente	Registro de entregables sin necesidad de lógica.
Handover Report	Parcialmente	Permite certificar la entrega final formalmente.
Transition Plan	No	Plan operativo sin ejecución contractual.
Stakeholder Matrix	No	Herramienta analítica sin lógica de ejecución.
Communication Matrix	No	Define canales, no requiere trazabilidad.
Roles & Responsibilities Matrix	Parcialmente	Útil para definir roles en contratos, no esencial.
Project Assumptions Log	Parcialmente	Registro de suposiciones puede justificar decisiones.
Project Constraints Log	Parcialmente	Podría documentar límites relevantes al contrato.
Project Dependencies Log	Parcialmente	Verificación de dependencias posible pero compleja.
Project Estimations	No	Datos estimados sin necesidad de inmutabilidad.
Project Metrics	Parcialmente	Métricas clave podrían auditarse, no críticas.
Project Status Report	Parcialmente	Informe validado públicamente posible, no necesario.
Kick-off Presentation	No	Presentación sin valor contractual.
Change Request Form	Muy recomendable	Permite flujos de revisión y aprobación claros.
Change Control Note	Sí	Complementario al *Change Log* para trazabilidad.

En conclusión, esta metodología de análisis proporciona un enfoque claro y sistemático para la selección estratégica de artefactos PM², garantizando que la implementación *blockchain* mediante contratos inteligentes esté plenamente alineada con el principio fundamental de PM²: gobernanza transparente, eficiente y segura del proyecto.

Capítulo 3 Implementación

En el capítulo anterior se expuso el análisis de viabilidad aplicado al conjunto completo de los 33 artefactos definidos por la metodología PM². A partir de ese estudio se identificaron cinco artefactos (clasificados como "muy recomendables" por su alto valor en términos de trazabilidad, automatización y control) como candidatos idóneos para una prueba de concepto basada en contratos inteligentes de Ethereum.

La elección se basó en tres criterios clave: viabilidad técnica, valor añadido por trazabilidad inmutable y potencial de automatización, aplicados de forma sistemática a los 33 artefactos definidos por PM².

Este capítulo describe, de manera detallada e individualizada, la propuesta de implantación *on-chain* (es decir, directamente sobre la cadena de bloques) de los artefactos *Project Charter*, *Change Log*, *Change Request Form*, *Deliverable Acceptance Form* y *Deliverable Acceptance Plan*.

Para cada uno se presenta la problemática que resuelve, la lógica de negocio trasladada a *blockchain* y los beneficios tangibles que introduce en el flujo de gestión del proyecto. El objetivo es demostrar que la tecnología de libro mayor distribuido puede transformar artefactos críticos de la metodología PM² en registros auto-ejecutables, verificables y resistentes a la alteración, reforzando así los principios de transparencia, responsabilidad y mejora continua que caracterizan a la gestión de proyectos moderna.

3.1 Implementación *Blockchain* del Artefacto *Project Charter*

El *Project Charter*, o Documento de Constitución del Proyecto, es uno de los artefactos fundamentales de inicio en la metodología PM². Tiene como función principal autorizar formalmente el proyecto y otorgar autoridad al Project Manager para utilizar los recursos organizativos asignados. Además, define de forma preliminar los objetivos, el alcance, los principales entregables, riesgos de alto nivel, interesados clave, y los criterios de éxito del proyecto.

Este artefacto cumple una función estratégica, ya que representa el acuerdo inicial entre el patrocinador y el equipo de proyecto. Sin embargo, en entornos tradicionales, el *Project Charter* suele gestionarse como un documento estático en formato Word o PDF, firmado digital o manualmente por los responsables, y almacenado en sistemas internos sin control automatizado sobre versiones, modificaciones o autorizaciones posteriores. Esta práctica presenta limitaciones significativas, especialmente en proyectos multicorporativos o con alta rotación de personal.

A continuación, se identifican y describen los principales desafíos que presenta el uso convencional del artefacto *Project Charter* cuando se gestiona mediante herramientas tradicionales. Estos problemas afectan a la integridad, trazabilidad, control de versiones y validación del documento, lo que puede generar riesgos operativos y estratégicos en fases tempranas del proyecto. La Tabla 3 resume estos retos clave.

Tabla 3. Desafíos del artefacto *Project Charter* tradicional. Elaboración propia con Microsoft Excel [71].

Reto habitual	Descripción del problema
Autenticidad de origen	Resulta difícil verificar si el documento fue efectivamente emitido por el patrocinador designado.
Trazabilidad de versiones	No siempre queda claro cuál es la versión vigente, especialmente si se generan en paralelo.
Aprobación y firma	Las firmas pueden ser simbólicas (un nombre escrito o una firma escaneada) y manipulables.
Inmutabilidad del acuerdo	Tras su firma, el documento puede ser modificado sin dejar rastro ni aviso formal.
Auditoría ex post	Comprobar quién firmó qué y cuándo suele requerir revisar correos, actas y carpetas locales.
Falta de estandarización	Cada organización puede redactarlo de forma distinta, lo que dificulta su validación externa.

3.1.1. Cómo *Blockchain* Elimina o Mitiga estos Problemas

La digitalización del *Project Charter* a través de tecnología *blockchain* permite superar varias de las limitaciones que presenta su gestión tradicional. A continuación, se detallan los mecanismos mediante los cuales esta transformación aporta mejoras sustanciales:

1. **Emisión verificable desde una autoridad legítima**: el contrato inteligente correspondiente está programado para aceptar únicamente transacciones originadas desde una dirección específica, vinculada criptográficamente al patrocinador del proyecto. Esto garantiza que la emisión del *Project Charter* solo pueda ser iniciada por quien realmente tiene la autoridad para hacerlo, evitando suplantaciones o accesos no autorizados.

2. **Gestión de versiones única e inmutable**: cada edición del *Project Charter* se registra como una entrada única en la *blockchain*, acompañada de su huella digital (*hash*). Si se requiere modificar o actualizar el contenido, debe emitirse una nueva versión que refiera explícitamente a la anterior. Este mecanismo permite mantener un historial perfectamente encadenado y consultable, eliminando riesgos de pérdida de trazabilidad o ediciones encubiertas.

3. **Firmas criptográficas seguras y verificables**: tanto el patrocinador como el Project Manager pueden aprobar el documento firmando con su clave privada. Estas firmas se registran como transacciones independientes, verificables en la cadena, lo que garantiza la autenticidad de las aprobaciones y elimina cualquier posibilidad de falsificación o ambigüedad en la validación.

4. **Control de acceso automatizado y preciso**: la lógica interna del contrato inteligente permite especificar exactamente qué direcciones están autorizadas para emitir, aprobar o consultar el artefacto. Esto refuerza la seguridad del documento, evitando accesos accidentales o

manipulaciones por parte de personal no autorizado, sin necesidad de sistemas externos de gestión de permisos.

5. **Sello de tiempo distribuido e irrefutable**: cada acción registrada en la *blockchain* incluye un sello de tiempo generado automáticamente por la red. Este *timestamp* es inalterable y aceptado por consenso, eliminando posibles disputas sobre la fecha o el momento exacto de aprobación del *Project Charter*.

6. **Verificación del contenido mediante *hash***: en lugar de almacenar el documento completo, la cadena conserva su *hash* criptográfico. Esto permite verificar que cualquier copia externa (por ejemplo, un PDF disponible en un repositorio institucional) sea exactamente la misma versión aprobada, sin necesidad de confiar en la integridad del sistema donde esté alojada.

3.1.2. Beneficios Adicionales Alineados con PM²

La implementación en la cadena (*on-chain*) del *Project Charter* no solo resuelve problemas funcionales, sino que también refuerza los valores fundamentales de la metodología PM². Algunos beneficios estratégicos que se derivan de esta solución son:

- **Transparencia interinstitucional asegurada** en proyectos colaborativos, especialmente en aquellos con financiación pública o multinivel, todas las partes pueden consultar la versión vigente del *Project Charter* de forma directa en la *blockchain*, sin depender de sistemas centralizados ni credenciales específicas.

- **Agilidad en auditorías y revisiones** gracias a la trazabilidad inherente de la *blockchain*, los auditores pueden comprobar que el procedimiento de aprobación se ha seguido correctamente, cuándo tuvo lugar y qué personas intervinieron, sin necesidad de solicitar archivos locales o registros manuales.

- **Reducción de ambigüedades y disputas contractuales** al estar definidas de forma codificada las reglas de aprobación y los firmantes válidos, desaparecen los márgenes de interpretación sobre la legitimidad del documento o las condiciones bajo las cuales fue validado.

- **Refuerzo de la confianza desde el inicio del ciclo de vida**, pues la existencia de un *Project Charter* inalterable, transparente y verificable desde su génesis establece una base sólida de gobernanza. Esto es especialmente relevante en contextos complejos, regulados o con alta rotación de personal, donde mantener la continuidad documental y la legitimidad del proceso resulta crítico.

A modo de recapitulación, se puede decir que la implementación *on-chain* (en la cadena) del *Project Charter* transforma este artefacto estratégico en un documento auto-verificable, inviolable y completamente trazable, elevando los estándares de gobernanza y seguridad desde el primer paso del ciclo de vida del proyecto. Esta solución refuerza los pilares de PM², permitiendo una gestión basada en confianza distribuida, especialmente útil en contextos regulados, colaborativos o con alta rotación de personal

3.2 Implementación *Blockchain* del Artefacto *Change Log*

La Bitácora de Cambios, o *Change Log*, es uno de los artefactos más relevantes dentro de la metodología PM², ya que permite registrar y dar seguimiento a todas las solicitudes de modificación significativas que puedan surgir a lo largo del ciclo de vida del proyecto. Este documento incluye habitualmente campos como el identificador del cambio, su descripción, la fecha de propuesta, el origen, los responsables implicados, su estado (propuesto, aprobado, rechazado o cerrado), así como las observaciones o decisiones tomadas al respecto. Gracias a esta información, se garantiza la trazabilidad y justificación formal de los desvíos respecto a la planificación inicial.

Sin embargo, en su implementación tradicional (mediante hojas de cálculo, formularios digitales o documentos compartidos) este artefacto presenta una serie de limitaciones estructurales y operativas. En particular, la exigencia de PM² en cuanto a la trazabilidad, el control de versiones, la validación de las decisiones adoptadas y la autenticidad de las firmas no siempre puede cumplirse de forma fiable con medios convencionales.

La Tabla 4 resume estos desafíos agrupados en seis categorías fundamentales, que corresponden a las filas de la tabla: integridad de datos, trazabilidad, no repudio, gobernanza y quórum, sello de tiempo confiable e integridad documental. En cada caso, se describe a la derecha (columna "Descripción del problema") la dificultad específica que suele surgir al gestionar el *Change Log* con herramientas ofimáticas tradicionales. Este análisis permite entender por qué es necesario replantear su implementación con tecnologías más robustas, como blockchain, que permiten cubrir dichas carencias de forma estructural y automatizada.

Tabla 4. Desafíos del artefacto *Change Log* tradicional. Elaboración propia con Microsoft Excel [71].

Reto habitual	Descripción del problema
Integridad de datos	Los campos (descripción, fecha, aprobadores) pueden editarse a posteriori sin dejar rastro, lo que abre la puerta a errores o manipulaciones.
Trazabilidad	Reconstruir la "historia" de un cambio exige cotejar correos, actas y versiones del fichero; el proceso es lento y propenso a lagunas.
No repudio	Las firmas suelen consistir en un nombre escrito o la cuenta de red del aprobador; resulta difícil demostrar que, efectivamente, esa persona aceptó la decisión.
Gobernanza y quórum	Corresponde al PM revisar que se haya alcanzado el número mínimo de aprobaciones y que sólo voten quienes tienen autoridad. Esto requiere control manual.
Sello de tiempo confiable	Dependiendo del servidor o la zona horaria, el momento exacto de la aprobación puede ser discutible.
Integridad documental	Adjunto al cambio se almacena un acta o un plano; si ese archivo se sustituye, la referencia en el registro no lo detecta.

3.2.1. Cómo *Blockchain* Elimina o Mitiga estos Problemas

1. **Inmutabilidad y sello de tiempo distribuido**. cada vez que un interesado propone, aprueba, rechaza o cierra un cambio, la acción se registra como transacción en la red. Ese registro incluye un *timestamp* consensuado por los nodos y no puede modificarse sin invalidar todo el historial. El debate sobre "qué fecha es la oficial" desaparece.

2. **Gobernanza automática**: las reglas de quién puede aprobar, cuántos votos se necesitan y qué plazo existe para emitirlos se programan directamente en el contrato. La cadena sólo acepta la transición de estado cuando se cumplen esos criterios, de forma que el Project Manager no tiene que comprobar manualmente el quórum ni vigilar intentos de voto no autorizado.

3. **No repudio basado en criptografía**: cada firma es una clave privada que controla una dirección única en la red. Cuando un aprobador emite su voto, la transacción queda matemáticamente ligada a su identidad digital, eliminando la ambigüedad de las firmas manuscritas o de las cuentas de red.

4. **Bitácora transparente y tiempo real**: todas las partes (PM, aprobadores, auditores) pueden consultar en un explorador de bloques el historial completo sin pedir acceso a un sistema corporativo. La trazabilidad no requiere rescatar correos guardados: los eventos *on-chain* muestran de manera ordenada quién hizo qué y cuándo.

5. **Integridad de documentos**: en lugar de almacenar el archivo adjunto, el registro conserva su huella digital (*hash*). Cualquier persona puede comprobar que el PDF o el plano que se consulta hoy es idéntico al que fue aprobado, simplemente recalculando el *hash*. Si alguien intentara reemplazar el documento, la discrepancia se detectaría al instante.

6. **Ciclos de mejora continua sin pérdida de histórico**: PM² contempla la posibilidad de reabrir un cambio cerrado si hay nueva información. En la versión *blockchain*, al reabrir se conserva el rastro completo de ciclos previos (propuesta inicial, aprobaciones, cierre) y se inicia una ronda fresca de votos. Así, la bitácora crece de forma encadenada sin sobrescribir datos.

3.2.2. Beneficios Adicionales Alineados con PM²

- **Métricas "vivas"**: en cualquier momento se conocen los números de cambios abiertos, aprobados o rechazados, lo que facilita los KPI de seguimiento del proyecto.

- **Auditoría externa simplificada**: un auditor puede validar la aplicación estricta del procedimiento simplemente consultando los eventos de la cadena, sin solicitar capturas de pantalla ni permisos de base de datos.

- **Reducción de costes operativos**: al automatizar quórum, control de roles y sellado horario, el equipo evita tareas administrativas y focos de error humano.

En conjunto, la implementación en la cadena (*on-chain*) convierte la Bitácora de Cambios PM² en un registro auto-ejecutable, verificable y resistente a la alteración, reforzando así los principios de transparencia, responsabilidad y trazabilidad que la metodología exige.

3.3 Implementación *Blockchain* del Artefacto *Change Request Form* (CRF)

El *Change Request Form* (CRF), o Formulario de Solicitud de Cambio, es el instrumento formal que la metodología PM² proporciona a los interesados para plantear modificaciones sobre parámetros clave del proyecto como el alcance, presupuesto, cronograma o calidad. Este documento actúa como punto de partida oficial para la evaluación y aprobación de un cambio, y debe contener información clara, estructurada y suficiente para facilitar la toma de decisiones informadas.

La Tabla 5 recoge los principales bloques de información que suelen encontrarse en un CRF típico, junto con ejemplos de campos representativos que los componen. Las filas representan dichas secciones clave (identificación, descripción, análisis, etc.), mientras que las columnas muestran ejemplos concretos que ilustran cómo se plasma esta información en la práctica. Esta estructura permite estandarizar la documentación y agilizar su interpretación por parte de los diferentes responsables del proceso de validación.

Tabla 5. Contenido de un Formulario de Solicitud de Cambio. Elaboración propia con Microsoft Excel [71].

Bloque de información habitual	Ejemplos de campos
Identificación de la petición	Nº de referencia, fecha de registro, autor/origen
Descripción detallada	Qué se quiere cambiar y por qué
Análisis de impacto	Estimación de coste, esfuerzo, plazos, riesgos, interdependencias
Clasificación y prioridad	Urgencia, severidad, tipo de cambio
Ruta de aprobación	Responsables que deben revisar: *Sponsor*, PM, comité de cambios
Decisiones y firmas	Aprobado/Rechazado, condiciones, fecha, comentarios

En contextos tradicionales, los CRF suelen gestionarse mediante herramientas como correos electrónicos, formularios web o documentos alojados en intranets corporativas. Aunque funcionales, estos métodos presentan importantes debilidades operativas cuando se trata de asegurar la trazabilidad, el control de versiones o el cumplimiento de los plazos establecidos por PM².

La Tabla 6 detalla los principales retos asociados a la tramitación convencional de los formularios de solicitud de cambio. Las filas representan distintos tipos de problemas detectados a lo largo del ciclo de vida de un CRF, desde la falta de control sobre las versiones del análisis de impacto, hasta la ausencia de mecanismos que aseguren el cumplimiento de los acuerdos de nivel de servicio (SLA). La segunda columna explica por qué cada uno de estos desafíos es crítico, y cómo puede comprometer la validez del proceso de decisión o generar riesgos de incoherencia documental.

Tabla 6. Desafíos del artefacto *Change Request Form* tradicional. Elaboración propia con Microsoft Excel [71].

Desafío específico en la tramitación de CRF	Por qué es crítico
Versionado del impacto	Durante la negociación, los valores de coste o plazo cambian y resulta difícil saber cuál fue la cifra finalmente aprobada.
Secuencia de revisiones	El orden (analista → *sponsor* → PM) no siempre se respeta; a veces se salta un revisor o se producen "firmas cruzadas".
SLA de evaluación	PM² fija plazos máximos para emitir un dictamen; si alguien se retrasa no existe un disparador automático.
Vinculación con anexos	El análisis de impacto suele adjuntar cronogramas y hojas de cálculo. Si alguno se altera, la decisión tomada queda desfasada.

3.3.1. Cómo la Lógica *On-Chain* Resuelve estos Puntos

1. **Instancia única y versión inmutable**. Cada CRF se crea como un registro autónomo en *blockchain*; cualquier modificación del análisis de impacto se almacena como una transacción adicional, quedando un "árbol" cronológico de revisiones. Nunca se sobre-escribe la cifra anterior, por lo que el auditor puede ver la evolución del coste estimado o del retraso calculado.

2. **Flujo de aprobación programado**. En lugar de confiar en que los participantes respeten la ruta manual, el contrato define qué rol debe firmar primero, segundo y tercero. La red sólo permite que la petición avance cuando la firma que llega corresponde al paso correcto. Se evita la aprobación fuera de secuencia.

3. **Control automático de plazos (SLA)**. El CRF incorpora un temporizador interno: si un responsable no firma dentro del plazo asignado, la solicitud se marca como "expirada" y se notifica al solicitante de cambio. Así se cumple la práctica PM² de tiempos de respuesta transparentes.

4. **Prueba criptográfica de autoría**. Las firmas quedan vinculadas a la clave privada de cada responsable. No es posible que otra persona "firme en nombre de" sin controlar esa clave, satisfaciendo los requisitos de no-repudio de la metodología.

5. **Integridad del análisis de impacto**. Cada anexo (por ejemplo, el cronograma actualizado) se carga fuera de la cadena y se referencia en el CRF mediante su *hash*. Cualquier alteración

posterior del documento produce un *hash* diferente y, por tanto, una discrepancia detectable al instante.

6. **Visibilidad en tiempo real para el *Change Log***. Una vez que el CRF queda aprobado, el contrato emite un evento que el módulo *Change Log* escucha y registra automáticamente. Se evita la doble carga de información y se garantiza la coherencia entre artefactos.

3.3.2. Beneficios Añadidos frente a la Práctica Convencional

- **Trazabilidad integral**: del primer borrador de impacto al último comentario del comité, todo es reconstruible con un explorador de bloques.
- **Transparencia de prioridades**: cualquiera puede ver qué CRF están pendientes, cuál es su urgencia y cuánto tiempo llevan atascados.
- **Auditoría simplificada**: el auditor ya no pide "la versión 4 del Excel de impacto"; sólo recalcula el *hash* y comprueba que coincide con el registrado.
- **Reducción de disputas**: al existir un sello de tiempo distribuido, las discusiones sobre "cuándo se aprobó realmente" desaparecen.
- **Alineación con PM²**: el contrato respeta el ciclo propuesto de análisis, decisión y actualización del *Change Log*, garantizando que el proceso se ejecute siempre del mismo modo, sin depender de la disciplina manual de los participantes.

En síntesis, trasladar el *Change Request Form* a *blockchain* convierte un flujo propenso a retrasos, confusiones de versión y dudas de autoría en un procedimiento auto-gestionado, verificable y seguro, reforzando los principios de transparencia, control y mejora continua que PM² promueve.

3.4 Implementación *Blockchain* del Artefacto *Deliverable Acceptance Form* (DAF)

El *Deliverable Acceptance Form* (DAF), o Formulario de Aceptación de Entregables, constituye en la metodología PM² el instrumento formal mediante el cual el cliente (o el *Project Owner*) valida que un entregable ha sido recibido conforme a los requisitos pactados y, por tanto, puede considerarse aceptado. Esta aceptación no solo afecta a la gestión de calidad, sino también al control de pagos, cumplimiento contractual y avance formal del proyecto.

La Tabla 7 muestra las secciones más comunes que integran un DAF, así como el tipo de información que recogen. Las filas representan los bloques funcionales habituales del documento, desde los datos identificativos del entregable hasta las observaciones y firmas. Las columnas describen ejemplos concretos de contenido que habitualmente se incluye en cada sección, reflejando el flujo de revisión y conformidad entre las partes involucradas.

Tabla 7. Contenido del Formulario de Aceptación de Entregables. Elaboración propia con Microsoft Excel [71].

Sección habitual en un DAF	Contenido
Identificación	Nombre y código del entregable, hito al que pertenece
Referencias contractuales	Especificación, criterios de aceptación, versión aprobada
Resultado de la revisión	Aceptado, aceptado con reservas, rechazado
Observaciones	Defectos menores, acciones correctivas, fecha límite de corrección
Firmas	Responsable del proveedor, responsable del cliente, fecha

La gestión tradicional del DAF enfrenta problemas frecuentes tales como:

- **Falta de unicidad de la versión**: el entregable puede volver a circular con correcciones y el equipo acaba con varios PDFs firmados, sin tener claro cuál es el "definitivo".

- **Riesgo de firmas incompletas**: a veces el proveedor firma y envía el DAF; el cliente imprime, firma, escanea… y el PDF final nunca vuelve al sistema central. El estado real del entregable resulta ambiguo.

- **Dependencia de repositorios cerrados**: las evidencias (pruebas funcionales, capturas, informes de calidad) se almacenan en carpetas internas. El auditor externo debe solicitar acceso y confiar en que no se haya alterado nada.

- **Seguimiento de reservas y correcciones**: si la aceptación es "con reservas", las correcciones se tramitan en cadenas de correo independientes. Puede perderse la conexión entre los defectos y su cierre.

3.4.1. Aportes de la Lógica *On-Chain*

la gestión tradicional de los DAFs suele estar sujeta a errores operativos y ambigüedades que afectan directamente a la transparencia, trazabilidad y fiabilidad de la aceptación de entregables. Estas limitaciones se agravan cuando los registros dependen de soportes físicos, intercambios por correo electrónico o almacenamiento en servidores internos no accesibles a todas las partes interesadas.

La Tabla 8 presenta una síntesis de cómo la incorporación de tecnología *blockchain* mejora sustancialmente cada uno de los requisitos clave definidos por PM² para la aceptación de entregables. La columna izquierda identifica los aspectos fundamentales del proceso (como la unicidad documental, la visibilidad de evidencias o el control de firmas). La columna derecha describe cómo un sistema basado en contratos inteligentes *on-chain* puede resolver dichos retos, aportando garantías de integridad, automatización de flujos y conectividad directa con otros artefactos del proyecto, como el *Change Log*.

Tabla 8. Mejoras del artefacto *Deliverable Acceptance Form* gracias a *blockchain*. .Elaboración propia con Microsoft Excel [71].

Requisito PM²	Mejora ofrecida por *blockchain*
Unicidad y trazabilidad de la versión aceptada	El contrato registra un *hash* del entregable aceptado. Cualquier modificación posterior cambia el *hash* y evidencia la diferencia.
Proceso de firma simultánea	El flujo *on-chain* exige que tanto proveedor como cliente firmen; sólo entonces el estado pasa a "Aceptado". La red garantiza que ninguna firma se pierda o se falsifique.
Visibilidad permanente de evidencias	Las pruebas de aceptación (logs, reportes) se suben a un almacenamiento descentralizado (p.ej. IPFS) y su *hash* se liga al DAF. El auditor puede validar la integridad sin pedir acceso interno.
Gestión de reservas	Si el cliente añade reservas, el contrato abre automáticamente una "lista de acciones correctivas" y fija una fecha límite. Cuando el proveedor sube la evidencia de corrección, la marca se actualiza y queda registrada la trazabilidad.
Sello temporal confiable	Las fechas de cada firma las determina la red, evitando disputas sobre cuándo se produjo la aceptación.
Notificación automática al *Change Log*	Una vez aceptado el entregable, el contrato emite un evento; si hubo reservas que implican cambios, el módulo de *Change Log* recibe la notificación e inicia automáticamente el flujo de cambio correspondiente.

3.4.2. **Beneficios Concretos para el Proyecto**

- **Seguridad jurídica**: la aceptación de un entregable queda anclada a la cadena, con firmas criptográficas de ambas partes, reforzando la validez contractual.
- **Reducción de disputas**: proveedor y cliente comparten una única fuente de verdad pública; no hay debate sobre "la última versión firmada".
- **Auditoría inmediata**: un tercero puede, en minutos, comprobar la integridad del entregable y de las evidencias de prueba sin acceder al entorno interno del proyecto.
- **Seguimiento automático de defectos**: las reservas generan tareas vinculadas dentro del propio contrato; su cierre queda registrado, evitando olvidos.
- **Coherencia con otros artefactos PM²**: la aceptación definitiva dispara eventos que sincronizan el *Change Log* y la *Product Baseline*, manteniendo todo el ecosistema documental actualizado sin intervención manual.

En definitiva, trasladar el *Deliverable Acceptance Form* a una infraestructura *blockchain* convierte la aprobación de entregables en un proceso seguro, transparente y auditable. Gracias a la inmutabilidad, la firma criptográfica y la integración automática con artefactos relacionados, se refuerzan los principios de trazabilidad, responsabilidad y mejora continua que la metodología PM² establece para la gestión de la calidad del proyecto.

3.5 Implementación *Blockchain* del Artefacto *Deliverables Acceptance Plan*

En el marco de la metodología PM², el *Deliverables Acceptance Plan* (DAP) o Plan de Aceptación de Entregables es un documento esencial para asegurar la calidad del proyecto. Define, para cada entregable, los criterios de aceptación aplicables, el método de verificación, los responsables de revisión y las evidencias necesarias. En la práctica, actúa como un contrato de calidad entre el equipo del proyecto y el cliente, siendo requisito indispensable antes de considerar un hito como completado.

La Tabla 9 recoge los elementos más habituales que componen un DAP, organizados por secciones funcionales junto con ejemplos de contenido típicos. Esta estructura permite alinear expectativas, garantizar trazabilidad y facilitar las auditorías sobre cumplimiento técnico y contractual.

Tabla 9. Contenido típico del Plan de Aceptación de Entregables. Elaboración propia con Microsoft Excel [71].

Contenido típico de un DAP	Ejemplo
Lista de entregables	Documento de requisitos v2.0, módulo software XYZ, manual de usuario…
Criterios de aceptación	"Debe pasar la prueba de rendimiento A con < 200 ms", "Cumple la norma ISO …"
Método de verificación	Revisión documental, ensayo de laboratorio, prueba funcional…
Responsable de verificación	Equipo QA interno, consultora externa, cliente…
Fecha/hito previsto	Sprint 5, Fase de Construcción, Final del proyecto…
Evidencias requeridas	Protocolos firmados, registros de pruebas, informes de auditoría

La gestión clásica del DAP presenta una serie de retos habituales:

- **Cambios silenciosos de criterio**: a mitad de proyecto se "ajusta" un criterio de aceptación sin dejar claro si todos los interesados lo han aprobado.
- **Seguimiento fragmentado**: cada equipo lleva su propia hoja de verificación; consolidar el estado global del hito es costoso y propenso a errores.
- **Fechas objetivo borrosas**: cuando se reprograma un hito, no siempre se actualiza el plan maestro, generándose desfases y reprocesos.

- **Falta de trazabilidad de evidencias**: las pruebas se guardan en servidores locales o herramientas de test; asociarlas inequívocamente al entregable resulta complejo.

- **Auditoría dividida**: ISO-9001 o CMMI exigen demostrar que los criterios fueron acordados y cumplidos; reunir la evidencia dispersa consume semanas.

3.5.1. Aportaciones de *Blockchain* al *Deliverables Acceptance Plan*

Cuando el DAP se gestiona de forma tradicional, en hojas de cálculo u otros documentos independientes, surgen limitaciones relacionadas con el control de versiones, la trazabilidad de evidencias y la visibilidad del progreso global del proyecto. En ese contexto, una implementación mediante *blockchain* aporta mejoras clave en términos de transparencia, automatización y verificación objetiva.

En la Tabla 10 se presenta una síntesis de los beneficios concretos que ofrece el uso de contratos inteligentes para este artefacto. Se muestra cómo las necesidades de PM² se ven reforzadas por las capacidades de la lógica *on-chain*, tales como marcas de tiempo inmutables, alarmas automáticas ante desviaciones y validaciones de rol. Esto convierte al DAP en un instrumento activo de control de calidad, más allá de su función documental.

Tabla 10. Mejoras del artefacto *Deliverables Acceptance Plan* gracias a *blockchain*. Elaboración propia con Microsoft Excel [71].

Necesidad PM²	Mejora *on-chain*
Versión controlada del criterio	Cada vez que un criterio cambia, la nueva versión se registra como transacción inmutable; el historial de criterios queda disponible públicamente.
Visión única del progreso	El DAP vive como *registro maestro* en *blockchain*. Cada verificación exitosa actualiza su estado; el porcentaje de avance se obtiene en tiempo real sin hojas de cálculo.
Reprogramación transparente	Al mover la fecha de aceptación de un entregable, el cambio queda sellado con marca temporal y motivo. Los involucrados reciben evento de notificación.
Anclaje de evidencias	El *hash* de cada informe de prueba o acta de revisión se vincula al ítem correspondiente. Cualquier alteración posterior del fichero es detectable instantáneamente.
Matriz de responsabilidades	El contrato valida que únicamente el rol designado (QA, auditor externo, cliente) pueda marcar un criterio como “verificado”.
Alarmas de retraso	Si la fecha comprometida llega y el criterio no está verificado, el sistema genera un evento de “alerta de desviación”, facilitando la acción correctiva temprana.

3.5.2. **Beneficios Alineados con la Calidad PM²**

- **Estatus "vivo" y fiable**: el Project Manager (PM) puede mostrar, en cualquier momento, un panel con el porcentaje de criterios cumplidos por entregable, sin depender de reportes manuales.

- **Auditoría acelerada**: el auditor sólo necesita la lista de *hashes* para cotejar la integridad de los informes, reduciendo drásticamente la preparación de la evidencia.

- **Reducción de riesgos de "aceptación tácita"**: si un criterio no se verifica a tiempo, el propio contrato lo hace visible y exige tratamiento (por ejemplo, pasar a *Change Request*).

- **Sincronización automática con el *Deliverable Acceptance Form***: cuando todos los criterios de un entregable están "en verde", se emite un evento que habilita al cliente a firmar el DAF, evitando pasos manuales intermedios.

- **Transparencia multi-parte**: proveedor, cliente y auditores comparten un único plan aceptado, imposible de modificar unilateralmente, fomentando confianza y reduciendo disputas.

La implementación *on-chain* del *Deliverables Acceptance Plan* convierte lo que tradicionalmente es un documento estático y proclive a desalineaciones en un sistema activo de gestión de la calidad, donde: los criterios de aceptación evolucionan de forma controlada, la verificación se registra de manera inmutable, las evidencias quedan indisolublemente ligadas al entregable y las desviaciones se detectan de forma temprana.

Todo ello refuerza la disciplina PM² de "calidad planificada y demostrada", elevando la madurez del proyecto y reduciendo costes de seguimiento y auditoría

Capítulo 4 Diseño e Implementación Técnica del Sistema basado en Contratos Inteligentes

Este capítulo describe el proceso completo de diseño e implementación técnica de la solución *blockchain* propuesta para la gestión de artefactos clave en proyectos PM². La programación se ha centrado en un contrato inteligente desarrollado en Solidity, desplegado sobre la red Ethereum, con el objetivo de digitalizar la Bitácora de Cambios (*Change Log*). Este artefacto fue seleccionado por su alto valor en términos de trazabilidad, automatización y gobernanza.

Para convertirlo en una herramienta funcional y útil para el director de proyectos, ha sido necesario definir con precisión la estructura de datos, los roles implicados, las reglas de aprobación y rechazo de cambios, los eventos que deben ser registrados y las funciones que permiten interactuar con el sistema.

Además del contrato inteligente, se ha desarrollado una interfaz web (*frontend*) para facilitar su uso a usuarios no técnicos, y se han realizado pruebas para validar la lógica implementada y garantizar su seguridad y fiabilidad.

Tanto el código fuente del contrato inteligente en Solidity como la aplicación *frontend* se encuentran disponibles públicamente en el siguiente repositorio de GitHub [72]:
https://github.com/danitoo8/MasterThesis [73].

4.1 Diseño del Contrato Inteligente

El objetivo principal del contrato inteligente es replicar, mediante programación sobre *blockchain*, la forma en que habitualmente se gestiona el artefacto Bitácora de Cambios (*Change Log*) en proyectos reales siguiendo la metodología PM². Para conseguirlo, fue necesario realizar primero un diseño detallado que estableciera claramente qué debe hacer el sistema, cómo debe hacerlo y quién puede realizar cada acción.

Por lo tanto, antes de comenzar a escribir código en Solidity, se identificaron los requisitos funcionales esenciales que debería cumplir el contrato inteligente: desde el control de quién puede acceder y ejecutar ciertas operaciones, hasta cómo registrar y seguir de forma segura cada solicitud de cambio a lo largo de todo su ciclo de vida.

A continuación, se presentan, uno a uno, los componentes clave definidos durante este proceso de diseño, empezando por el control de acceso y la seguridad basada en roles.

4.1.1. Control de Roles y Seguridad de Acceso

Una de las primeras cuestiones fundamentales que se abordó en el diseño fue determinar quién tendría permiso para realizar cada operación en el sistema. Para solucionar este problema, se utilizó un modelo estándar llamado AccessControl, proporcionado por la librería OpenZeppelin. Este modelo permite gestionar fácilmente roles específicos que limitan el acceso y uso de funciones críticas dentro del contrato inteligente.

Concretamente, se definieron tres roles principales que reflejan claramente las responsabilidades típicas dentro de la gestión de cambios en proyectos reales:

- DEFAULT_ADMIN_ROLE: es el administrador general, responsable de asignar o revocar otros roles dentro del sistema. Este rol tiene el máximo nivel de autoridad y garantiza la capacidad de gestionar usuarios en caso necesario.

- PROJECT_MANAGER_ROLE: representa al director o gestor de proyecto, quien tiene la autoridad exclusiva para marcar como cerrados aquellos cambios que previamente han sido aprobados por los usuarios correspondientes.

- CHANGE_APPROVER_ROLE: es el conjunto de usuarios autorizados para votar a favor o en contra de los cambios propuestos. Este rol se otorga a múltiples personas para asegurar que las decisiones sean tomadas de forma colaborativa y transparente.

La asignación inicial de estos roles ocurre automáticamente en el momento de desplegar el contrato inteligente en la *blockchain*. Además, en este proceso inicial, se verifica que exista un número suficiente de aprobadores para cumplir siempre con el mínimo necesario de votos establecido en la configuración del sistema. Esto evita posibles errores o bloqueos en la gestión de cambios una vez que el sistema esté operativo.

```
// ——————————————
// Constructor
// ——————————————
/**
 * @param initialApprovers Direcciones iniciales con rol de aprobador
 * @param _requiredApprovals Numero minimo de votos para aprobar
 * @param _approvalPeriodDays    Periodo para votar (en dias)
 */
constructor(
    address[] memory initialApprovers,
    uint256 _requiredApprovals,
    uint256 _approvalPeriodDays
) {
    if (initialApprovers.length < _requiredApprovals) revert AprobadoresInsuficientes();

    // Owner = administrador y gestor de proyecto
    _grantRole(DEFAULT_ADMIN_ROLE, msg.sender);
    _grantRole(PROJECT_MANAGER_ROLE, msg.sender);

    requiredApprovals = _requiredApprovals;
    approvalPeriod    = _approvalPeriodDays * 1 days;    infinite gas 2631000 gas

    // Asignar rol de aprobador
    for (uint256 i = 0; i < initialApprovers.length; ) {
        _grantRole(CHANGE_APPROVER_ROLE, initialApprovers[i]);
        unchecked { ++i; }
    }
}
```

Figura 8. Código correspondiente al constructor en el contrato inteligente desarrollado. Elaboración propia en Remix IDE [69].

4.1.2. **Parámetros de Configuración**

Una vez definido quiénes pueden interactuar con el contrato inteligente, era necesario configurar ciertas reglas básicas del funcionamiento del sistema. Estos parámetros se deciden en el momento del despliegue del contrato en la *blockchain*, y permanecen constantes durante toda la vida del proyecto para evitar alteraciones inesperadas o malintencionadas posteriores.

En particular, se han definido dos parámetros críticos:

- requiredApprovals: este parámetro indica el número mínimo de aprobaciones que un cambio propuesto debe recibir para ser considerado oficialmente aprobado dentro del sistema. Este valor asegura que las decisiones sean tomadas por consenso y no de forma unilateral.

- approvalPeriod: se refiere al tiempo máximo (en días) disponible para que los aprobadores emitan su voto sobre cada cambio propuesto. Si no se alcanza el número requerido de votos en este plazo, el cambio no puede avanzar y debe gestionarse nuevamente.

Ambos parámetros se establecen como constantes inmutables (*immutable*) dentro del contrato inteligente, garantizando así eficiencia operativa (ahorrando gas en la red Ethereum) y asegurando transparencia y previsibilidad en la ejecución de las reglas del sistema.

4.1.3. **Estructura de Datos y Ciclo de Vida**

Para gestionar de forma ordenada, transparente y eficiente cada solicitud de cambio registrada en la *blockchain*, se ha diseñado una estructura de datos específica denominada ChangeEntry. Esta estructura recoge y organiza toda la información relevante relacionada con un cambio desde su propuesta inicial hasta su resolución definitiva. La estructura ChangeEntry almacena los siguientes datos esenciales:

- ID: un identificador único que distingue cada solicitud de cambio.

- Descripción: texto claro que explica qué cambio se está solicitando.

- *Hash* del documento de respaldo: referencia criptográfica al documento justificativo almacenado fuera de la *blockchain* (por ejemplo, en IPFS). Esto permite asegurar la integridad y autenticidad del documento original sin necesidad de guardarlo directamente en la *blockchain*.

- Autor: dirección del usuario que inicialmente propuso el cambio.

- Estado actual: situación actual del cambio dentro del flujo (por ejemplo, propuesto, aprobado, rechazado, cerrado, etc.).

- *Timestamp*: fecha y hora exactas en que se registró o modificó por última vez el cambio.

- Aprobadores y comentarios: registro de quiénes han votado sobre el cambio y sus respectivos comentarios.

- Historial de reaperturas: cuando un cambio ya cerrado debe reconsiderarse, aquí se guarda el motivo y fecha de cada reapertura.

Cada solicitud de cambio pasa por diferentes estados (véase figura 9), claramente definidos en un ciclo de vida que refleja las fases reales en la gestión de proyectos bajo PM².

```
/// @notice Estado de cada solicitud de cambio
enum Status { Proposed, Approved, Rejected, Closed }
```

Figura 9. Fragmento de código correspondiente a los estados de cambio. Elaboración propia en Remix IDE [74].

Además, para ofrecer una visión más precisa sobre cómo interactúan estos estados, la figura 10 muestra el diagrama de flujo completo, indicando de manera explícita cómo se transiciona de un estado a otro según las acciones realizadas por los usuarios autorizados.

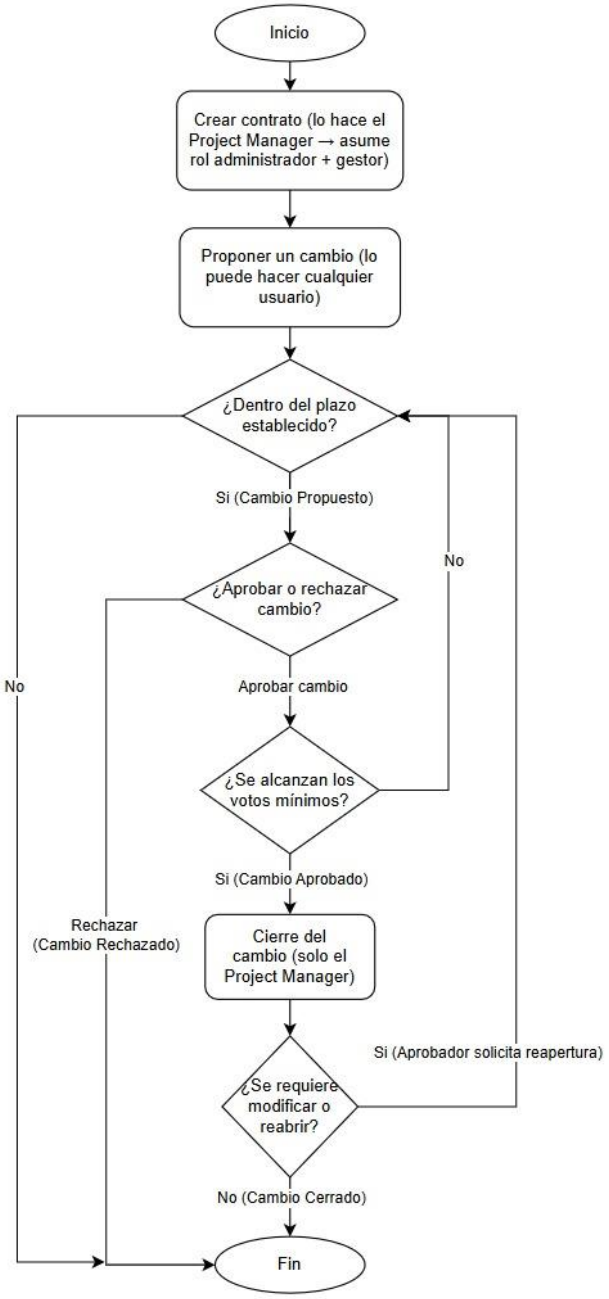

Figura 10. Diagrama de flujo de estados del cambio. Elaboración propia en diagrams.net [70].

4.1.4. **Flujo de Trabajo y Funciones de Negocio**

El contrato inteligente no solo almacena información estática, sino que además implementa un flujo de trabajo claramente definido, que simula fielmente la gestión real de solicitudes de cambio según la metodología PM². Este flujo incluye diversas acciones y decisiones, cada una controlada por funciones específicas que pueden invocarse según los permisos asignados a cada rol dentro del sistema. A continuación, se explica paso a paso cómo funciona este flujo de trabajo:

- Propuesta de Cambio (proposeChange): cualquier usuario autorizado puede iniciar una solicitud de cambio, proporcionando una breve descripción y el *hash* (huella digital) del documento que respalda la propuesta. Cuando se realiza esta acción, el sistema registra automáticamente la fecha de la propuesta y emite un evento llamado ChangeProposed, que queda permanentemente registrado en la *blockchain*.

- Aprobación (approveChange): los usuarios asignados con el rol de aprobador pueden votar positivamente sobre un cambio propuesto. Cada aprobación incluye un comentario justificativo. Cuando el número total de aprobaciones alcanza el mínimo establecido en la configuración, el cambio se marca automáticamente como aprobado (*Approved*) y se emite el evento ChangeApproved, notificando esta decisión a todos los interesados.

- Rechazo (rejectChange): de manera similar a la aprobación, los aprobadores también tienen la capacidad de votar negativamente sobre una propuesta. Cuando ocurre un rechazo, el cambio se marca inmediatamente como rechazado (*Rejected*), y se registra el evento ChangeRejected. Cada aprobador puede emitir solo un voto por cambio, garantizando que las decisiones sean claras y definitivas.

- Cierre (closeChange): una vez que un cambio ha sido aprobado formalmente, únicamente el usuario con el rol de Project Manager tiene autoridad para cerrarlo definitivamente, pasando al estado cerrado (*Closed*). Esta operación emite un evento ChangeClosed, finalizando oficialmente el proceso de aprobación.

- Reapertura (reopenChange): en situaciones en que un cambio previamente cerrado necesita reconsiderarse (por ejemplo, debido a nuevos hallazgos o requisitos adicionales), cualquier aprobador puede reabrirlo. Al hacerlo, se borran las aprobaciones anteriores para garantizar un nuevo proceso limpio, se actualiza la fecha y hora del evento, y se genera un evento ChangeReopened, que reinicia el ciclo de aprobación desde cero.

4.1.5. **Trazabilidad y Eventos**

Una ventaja fundamental del uso de blockchain es la capacidad de garantizar trazabilidad total y verificable de cada acción realizada en el sistema. Para aprovechar plenamente esta característica, se han definido eventos específicos que registran de forma inmutable y transparente todas las operaciones clave efectuadas en el contrato inteligente:

- ChangeProposed (cambio propuesto)

- ChangeApprovalReceived (aprobación individual recibida)

- ChangeRejected (cambio rechazado)

- ChangeApproved (cambio aprobado formalmente)
- ChangeClosed (cambio cerrado definitivamente)
- ChangeReopened (cambio reabierto)
- ChangeQueried (consulta del estado del cambio)

Estos eventos están claramente indexados, lo que facilita enormemente su localización y consulta a través de exploradores públicos de *blockchain* como Etherscan. Esto permite una auditoría externa sencilla y pasiva del historial completo de cambios, sin necesidad de mantener un almacenamiento externo adicional [75].

4.1.6. **Consultas y Funciones de Lectura**

Además de gestionar acciones, es fundamental que los usuarios puedan consultar fácilmente el estado actual e histórico de los cambios propuestos. Para esto, el contrato inteligente expone funciones específicas diseñadas para consultar datos sin coste adicional en gas (funciones *view*). Estas consultas permiten obtener información rápida y precisa sobre cualquier aspecto relevante del sistema, incluyendo:

- getChange(id): proporciona todos los detalles almacenados para una solicitud de cambio específica.
- getAllChangeIds(): devuelve una lista completa con todos los identificadores de cambios registrados.
- getUserChangeIds(address): permite listar todos los cambios propuestos por un usuario específico, facilitando así auditorías o seguimiento individualizado.
- countChangesByStatus(status) y listChangesByStatus(status): ofrecen estadísticas y listados filtrados según el estado actual del cambio (por ejemplo, aprobados, pendientes o rechazados).

Adicionalmente, existe una función especial queryChange() que, además de retornar el cambio solicitado, emite un evento de consulta denominado ChangeQueried, proporcionando así un registro adicional que refuerza la transparencia del sistema al aportar trazabilidad pasiva.

4.1.7. **Validaciones y Seguridad**

Finalmente, para asegurar que el contrato inteligente opere siempre de manera segura y según las reglas establecidas, se han implementado diversas validaciones de seguridad en todas sus funciones críticas. Estas validaciones tienen como objetivo evitar usos indebidos o incorrectos del sistema, protegiendo así la integridad del proceso y evitando errores comunes. Entre las medidas de seguridad implementadas destacan:

- Restricción estricta por rol (onlyRole(...)): limita la ejecución de funciones críticas únicamente a usuarios autorizados según su rol.

- Errores personalizados: en lugar de utilizar mensajes estándar (*require*), el contrato utiliza errores personalizados específicos (error) que ahorran gas y proporcionan mensajes más claros y detallados sobre la razón del fallo.

- Prevención del doble voto (YaVoto): impide que un aprobador pueda votar más de una vez sobre el mismo cambio.

- Control del plazo de votación (FueraDePlazo): garantiza que los votos emitidos después del periodo establecido no sean aceptados.

- Reapertura solo permitida sobre cambios cerrados (SoloClosed): evita que se puedan reabrir cambios que aún no hayan sido cerrados formalmente.

Estas validaciones aseguran que cada operación realizada sobre el contrato inteligente siga estrictamente las reglas establecidas y previenen intentos de fraude o errores operativos involuntarios.

4.2 Aplicación Frontend: Interfaz de Usuario para la Bitácora de Cambios

La solución incluye el desarrollo de una aplicación web basada en React [76] que actúa como interfaz visual para el contrato inteligente ChangeLog.sol, desplegado en la red Ethereum de pruebas Sepolia [77]. Su objetivo es proporcionar una forma accesible, segura y clara de interactuar con el sistema de gestión de cambios siguiendo la metodología PM².
La interfaz ha sido diseñada con énfasis en usabilidad, trazabilidad y seguridad de operación, y se comunica directamente con la *blockchain* mediante la librería ethers.js, usando MetaMask [78] como proveedor.

4.2.1. **Arquitectura General de la Aplicación**

La aplicación se construyó con React [76] utilizando Create React App, y se estructura de forma modular para facilitar el mantenimiento y la legibilidad del código. Los archivos clave son:

- **Variables de entorno (.env)**: contiene configuraciones sensibles y específicas del entorno, como:
    - La dirección del contrato inteligente desplegado, para apuntar correctamente al *backend blockchain*.
    - El chain ID de la red de pruebas (Sepolia [77]) para validar la conexión con MetaMask [78].

  Esta separación permite mantener el código genérico y reutilizable entre entornos de desarrollo y producción.

- **Interfaz del contrato (abi.js)**: incluye el ABI (*Application Binary Interface*) generado tras compilar el contrato. Esta estructura es esencial para que la aplicación pueda invocar funciones, escuchar eventos y recuperar datos desde la *blockchain* de forma estandarizada.

- **Estilos y diseño (App.css)**: define los estilos visuales de todos los elementos de la aplicación. Entre los aspectos destacados:

    - Diseño limpio y adaptable a distintos dispositivos.

  - Tarjetas para visualizar cambios, subcomponentes históricos y formularios.
  - Botones y formularios claramente identificables, con mensajes visuales y estilos accesibles.
  - Banners de error para mostrar información crítica de manera clara y no intrusiva.

- **Lógica y renderizado principal (App.js)**: este componente implementa la lógica central del frontend, incluyendo:

  - Conexión segura con MetaMask [78] y verificación automática de la red.
  - Consulta inicial de la configuración del contrato (votos requeridos, duración de aprobaciones).
  - Carga dinámica de cambios registrados y visualización de su estado y detalles.
  - Ejecución de acciones permitidas al usuario (proponer, votar, cerrar, reabrir).
  - Gestión de errores, validaciones y recarga automática del estado tras cada transacción.

4.2.2. **Funcionalidades Destacadas**

- **Conexión con MetaMask y detección de red**: la aplicación solicita al usuario que conecte su cartera MetaMask y verifica que esté operando en la red Sepolia. Si la red es incorrecta, se muestra un mensaje de advertencia y se bloquean las funciones hasta corregirlo.

- **Propuesta de cambios**: los usuarios pueden registrar nuevas solicitudes de cambio mediante un formulario que exige descripción y *hash* del documento justificativo. Una vez propuesto, el cambio se almacena *on-chain* y queda disponible para aprobación.

- **Aprobación y rechazo**: usuarios con rol de aprobador pueden emitir su voto positivo o negativo sobre cambios pendientes. Cada acción debe ir acompañada de un comentario justificativo y queda registrada en el historial como evento.

- **Cierre y reapertura**: una vez aprobado, un cambio puede ser cerrado por el gestor de proyecto. Si fuese necesario, los aprobadores pueden reabrir el cambio, iniciando una nueva ronda de validación. Todos los cambios de estado quedan reflejados en la interfaz, junto con los comentarios y los usuarios que los ejecutaron.

- **Timeline/histórico de eventos**: la aplicación recupera todos los eventos emitidos por el contrato para cada cambio y construye un historial visual ordenado cronológicamente. Esto permite auditar el ciclo completo de cada solicitud, incluyendo aprobaciones, rechazos, cierres y reaperturas.

- **Filtros y estadísticas**: el usuario dispone de filtros para visualizar cambios según su estado (propuesto, aprobado, rechazado, cerrado) o por dirección de usuario. También se presentan métricas básicas con el total de cambios en cada estado, ofreciendo una visión rápida del estado general del sistema.

- **Manejo de errores y validaciones**: la interfaz traduce los errores técnicos del contrato en mensajes amigables para el usuario. Por ejemplo, si se intenta votar dos veces o fuera de plazo, se muestra una alerta clara explicando el motivo. Esto mejora significativamente la experiencia de usuario, incluso ante errores esperados.

- **Experiencia de usuario y buenas prácticas:** el desarrollo se centró en asegurar una experiencia fluida, clara y libre de fricciones. Entre las buenas prácticas aplicadas destacan:

- Renderizado eficiente mediante useEffect y useCallback, evitando recargas innecesarias.
- Accesibilidad visual con botones claros, textos legibles, y navegación sencilla.
- Control de estados de la interfaz que se actualiza automáticamente tras cada interacción.
- Retroalimentación inmediata en cada operación (confirmación, error, actualización de pantalla).
- Separación lógica del código para mantener claridad y extensibilidad futura.

4.3 Verificación y Pruebas del Sistema

Para garantizar la fiabilidad y robustez del sistema, se ha llevado a cabo un proceso completo de verificación del funcionamiento tanto del contrato inteligente como de la interfaz web. Las pruebas incluyen escenarios positivos y negativos, interacción entre usuarios con distintos roles, y validación de los eventos y restricciones definidos en la lógica del contrato.

La verificación se realizó en la red de pruebas Sepolia [77], utilizando cuentas de test en MetaMask [78] y transacciones *on-chain* monitorizadas con Remix [74] y Etherscan [75]. La interacción se llevó a cabo a través de la interfaz desarrollada en React [76].

4.3.1. **Despliegue del Contrato en Remix IDE**

Antes de realizar las pruebas funcionales, es necesario desplegar el contrato inteligente ChangeLog.sol en una red pública de pruebas para que pudiera ser utilizado desde la interfaz web. Como se ha venido mencionando, para este fin, se utilizó el entorno de desarrollo Remix IDE, una herramienta en línea ampliamente utilizada para compilar, desplegar y depurar contratos inteligentes en Solidity. Para llevar a cabo el despliegue, se necesita tener cuenta de MetaMask configurada con la red Sepolia, poseer fondos de prueba (ETH Sepolia) en la cuenta para cubrir el coste de gas y definir los siguientes parámetros del constructor:

- initialApprovers: array de direcciones que tendrán rol de aprobador. Deben ser direcciones válidas de cuentas MetaMask disponibles. Por ejemplo: [“0x6bbEe96f820bef6Df57e39696Dc087d1344777D6”,” 0x91fcD2018A107D09584387ea4FfcB644f616B215”,” 0xeABDb47F9D87591bEAa7c22049f41bE88408F6fD”]
- requiredApprovals: número mínimo de votos necesarios para aprobar un cambio (por ejemplo, 2).
- approvalPeriodDays: duración de la ventana de votación (por ejemplo, 3 días).

En la Figura 11 se muestra el mensaje de confirmación del despliegue exitoso desde Remix. La Figura 12 detalla la interacción en MetaMask, donde el usuario revisa los parámetros y confirma la operación, validando el uso de la cuenta y los recursos asignados. Finalmente, en la Figura 13 se ilustra cómo la transacción correspondiente queda registrada en Etherscan, asegurando la trazabilidad del proceso de despliegue y permitiendo su posterior auditoría pública.

Figura 11. Despliegue satisfactorio del contrato inteligente. Visualizado en Remix IDE [74].

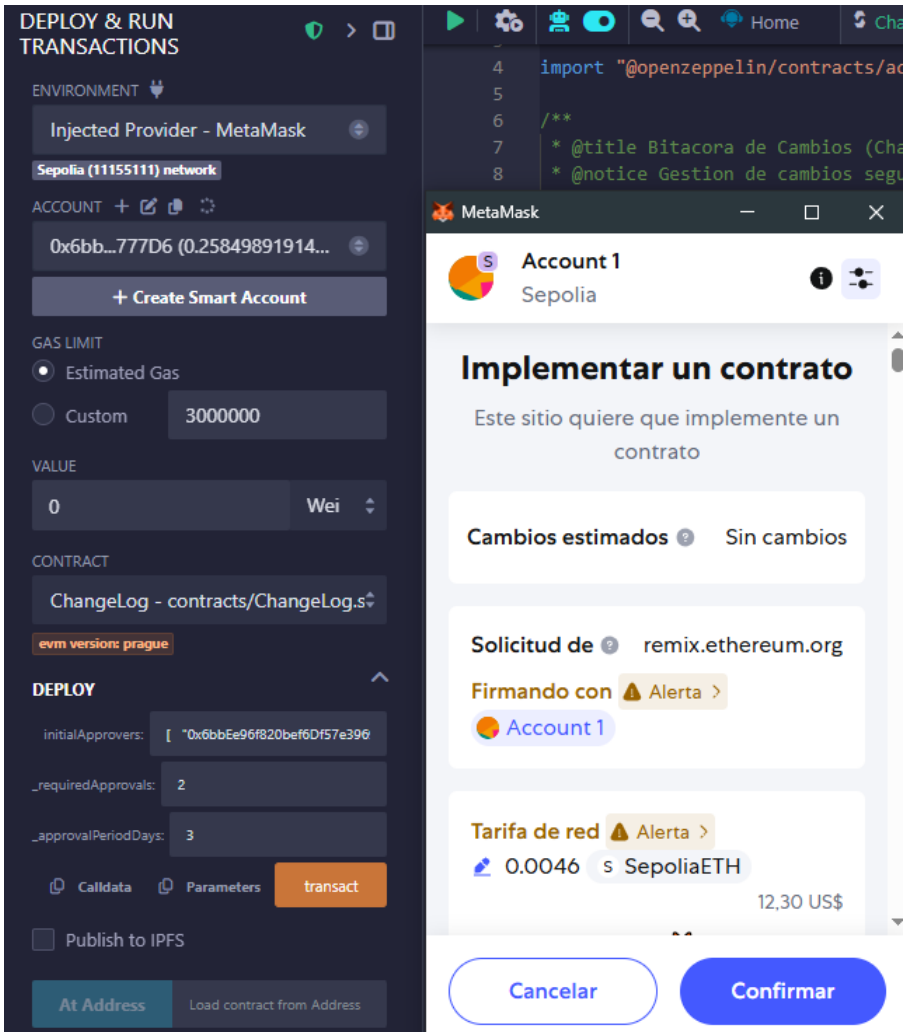

Figura 12. Definición de parámetros y confirmación de la transacción en MetaMask [78] para el despliegue del contrato inteligente desde Remix IDE [74].

Una vez desplegado, se obtiene la dirección del contrato en Remix [74]. Esta dirección debe copiarse y colocarse en el archivo. env de la aplicación React [76] como:

REACT_APP_CONTRACT_ADDRESS=0xaF4b6daA9Ed57243B4F373886D94041e62c5eA94
REACT_APP_CHAIN_ID=11155111

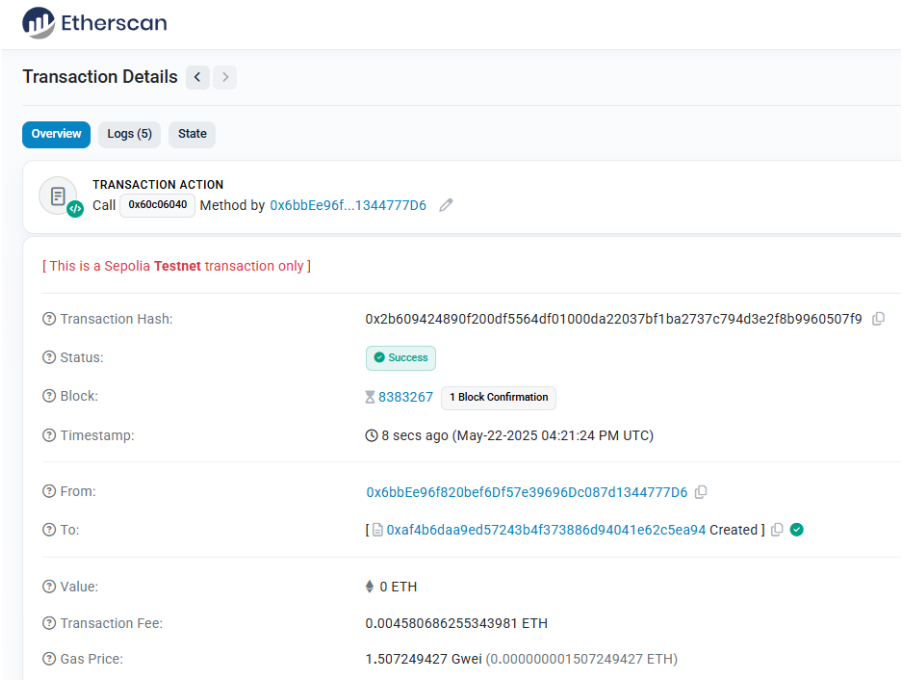

Figura 13.Visualización en Etherscan [75] de la transacción correspondiente al despliegue del contrrato inteligente.

A continuación, se lanza el servidor de desarrollo desde Git Bash [79], para encontrarnos con la interfaz que solicita conectarse con la billetera de MetaMask [78]. Véase Figura 14.

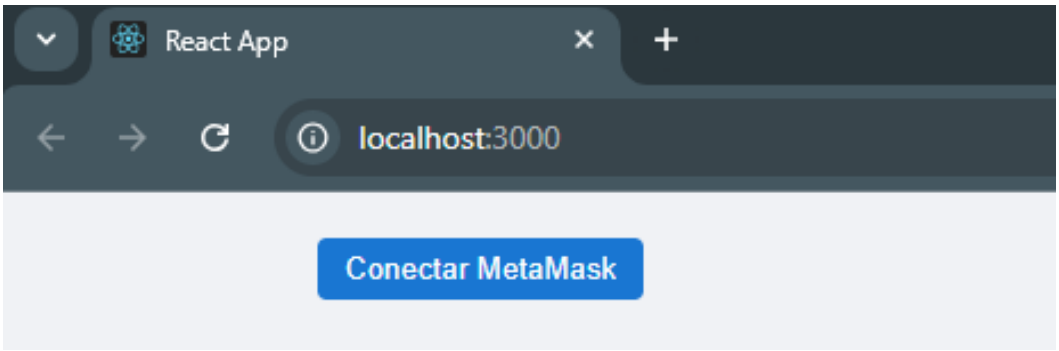

Figura 14. Servidor lanzado [76] listo para conectarse con la billetera MetaMask.

Una vez conectado correctamente, se muestra la cuenta activa, junto con los parámetros clave del sistema: votos requeridos y plazo de aprobación. Véase Figura 15.

Figura 15. Vista inicial de la Bitácora de Cambios [73].

4.3.2. **Propuesta de Cambio**

Cualquier usuario conectado puede proponer un nuevo cambio, proporcionando una descripción y el *hash* del documento justificativo. La propuesta queda almacenada en la *blockchain* y se emite un evento ChangeProposed.

Para proponer un cambio, basta con rellenar los campos solicitados (ver Figura 16), por ejemplo:

- **Descripción**: "Actualizar logo institucional en todos los documentos oficiales."
- ***Hash* documento**: 0x6fbc8b6e072fc399e8654c6a1be0df2e6e6b7b949c79cf7cb2eb55b61f8c221f

Este *hash* corresponde al documento PDF llamado: CambioLogo2025.pdf (simulado como si estuviera firmado o almacenado en IPFS) y propuesto por: 0x6bbEe96f820bef6Df57e39696Dc087d1344777D6.

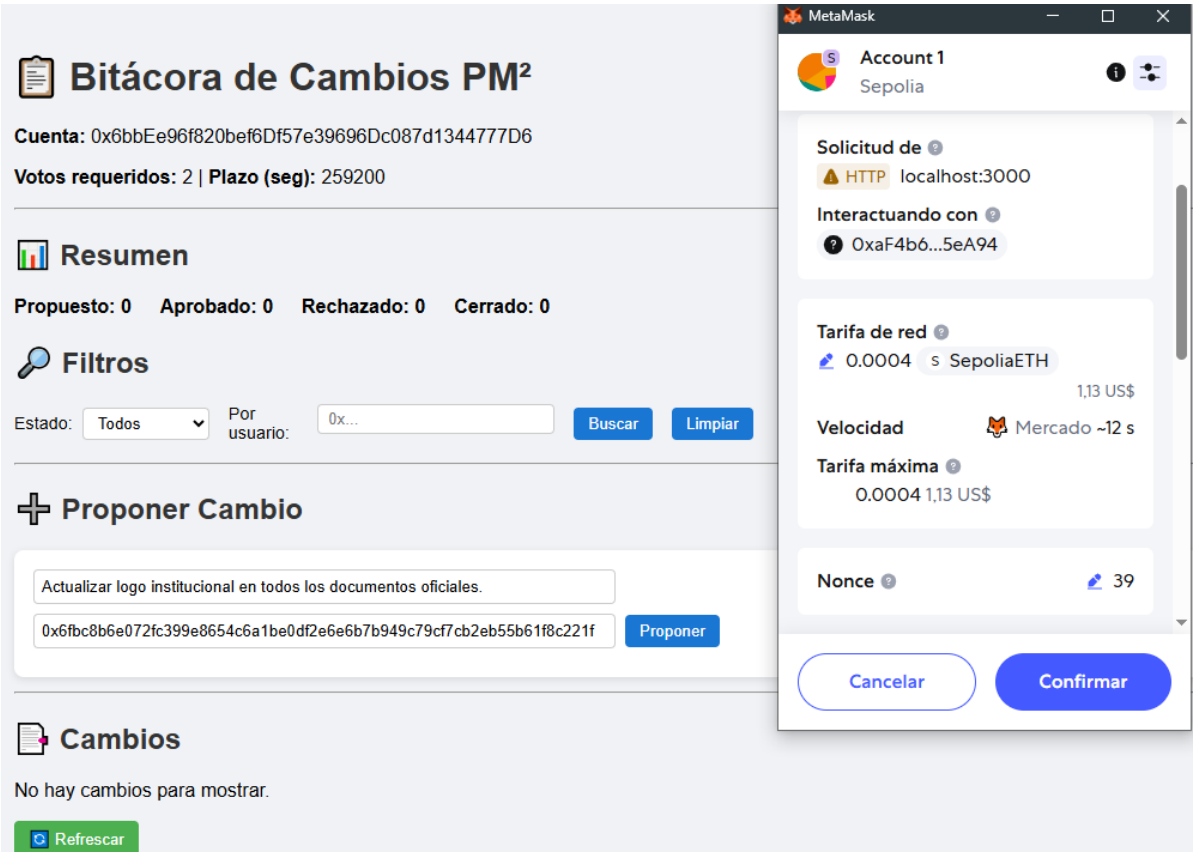

Figura 16.Cambio propuesto y solicitud de confirmación de la transacción [73].

Una vez confirmada la transacción para la solicitud de cambio, la interfaz de usuario mostrará una nueva entrada que incluye un identificador único (ID), el estado actual del cambio, una descripción del motivo de la solicitud, el *hash* del documento justificativo asociado (generado fuera de la cadena), la dirección Ethereum del usuario que propuso el cambio y la fecha y hora de registro. Además, se presentan los botones de acción correspondientes a los usuarios con rol de aprobador, quienes pueden emitir su voto (aprobación o rechazo) directamente desde la interfaz conectada con MetaMask. Véase Figura 17.

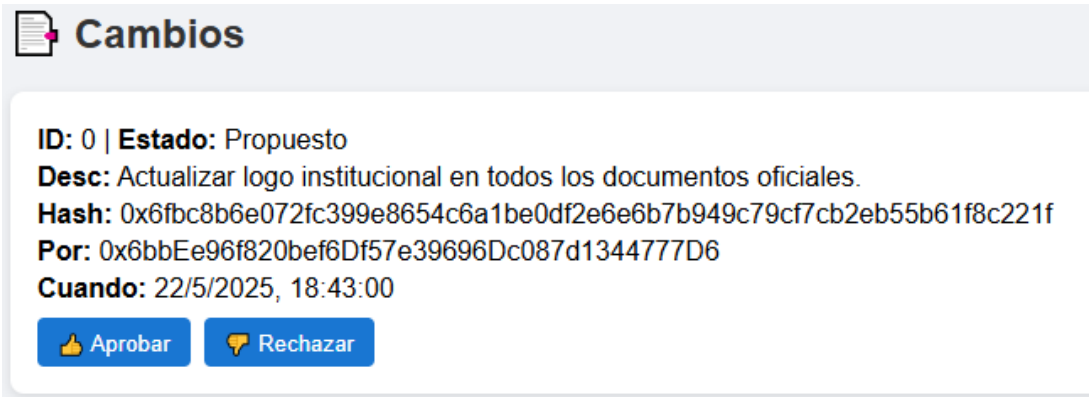

Figura 17. Registro del cambio propuesto [73].

4.3.3. **Aprobación de un Cambio**

El proceso de aprobación representa una de las acciones más relevantes dentro del ciclo de vida de una solicitud de cambio en el sistema desarrollado. Los usuarios que han sido asignados con el rol de aprobador pueden emitir su voto favorable sobre cualquier cambio propuesto. Cada voto requiere obligatoriamente un comentario justificativo, que se almacena junto a la marca de tiempo exacta de la transacción. Esta acción genera un evento en la *blockchain* denominado ChangeApprovalReceived, que garantiza la trazabilidad de la decisión tomada y la identidad digital del firmante.

Una vez alcanzado el umbral mínimo de aprobaciones establecido durante la configuración del contrato inteligente, se activa automáticamente el evento ChangeApproved, y el estado del cambio se actualiza de forma irreversible a "aprobado". Esta transición queda reflejada tanto en la interfaz web como en el historial *on-chain*, permitiendo verificar en cualquier momento quién aprobó, cuándo lo hizo y con qué observaciones.

Las figuras siguientes ilustran el flujo progresivo del proceso de aprobación dentro del sistema implementado.

En la Figura 18, se observa el momento en que un primer usuario con rol de aprobador emite su voto positivo sobre una solicitud de cambio. Esta aprobación queda registrada en el historial, junto con el comentario del usuario y la marca de tiempo correspondiente. En este punto, el cambio aún se encuentra en estado “Propuesto”, ya que no se ha alcanzado el número mínimo de aprobaciones requeridas para su validación definitiva.

Figura 18. Registro de la primera aprobación sobre una solicitud de cambio [73].

Posteriormente, como se muestra en la Figura 19, un segundo aprobador emite su voto favorable. Esta segunda confirmación activa automáticamente la lógica del contrato inteligente, que emite el evento ChangeApproved y actualiza el estado del cambio a “aprobado”. Esta transición de estado demuestra el funcionamiento efectivo del sistema de gobernanza distribuida basado en contratos inteligentes.

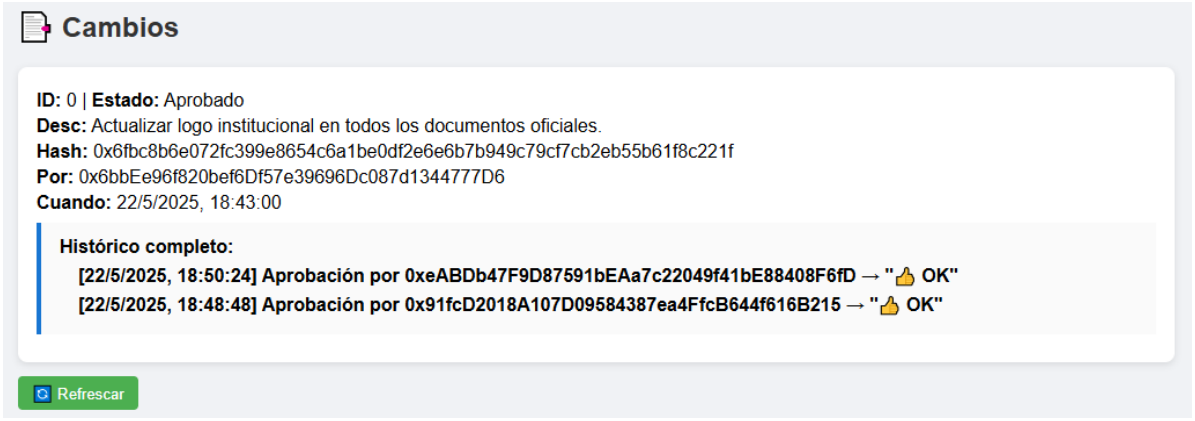

Figura 19. Segunda aprobación registrada. El cambio pasa automáticamente a estado "aprobado" [73].

4.3.4. **Rechazo de un Cambio**

Además de aprobar solicitudes, los usuarios con rol de aprobador también pueden emitir un voto negativo cuando consideren que un cambio no debe implementarse. En este caso, el sistema registra el rechazo de forma inmediata, actualizando el estado del cambio a "rechazado" sin necesidad de alcanzar un umbral mínimo de votos.

Este tipo de acción genera automáticamente el evento ChangeRejected, que queda registrado en la *blockchain* junto con la identidad del aprobador, el comentario justificativo y la marca de tiempo. Esta trazabilidad garantiza que cada decisión quede documentada de forma segura, permitiendo auditorías posteriores sin depender de sistemas externos.

En la Figura 20, se muestra un ejemplo de este escenario. El cambio propuesto fue rechazado por uno de los aprobadores, y dicha acción quedó reflejada en el historial del sistema, incluyendo su comentario y firma digital.

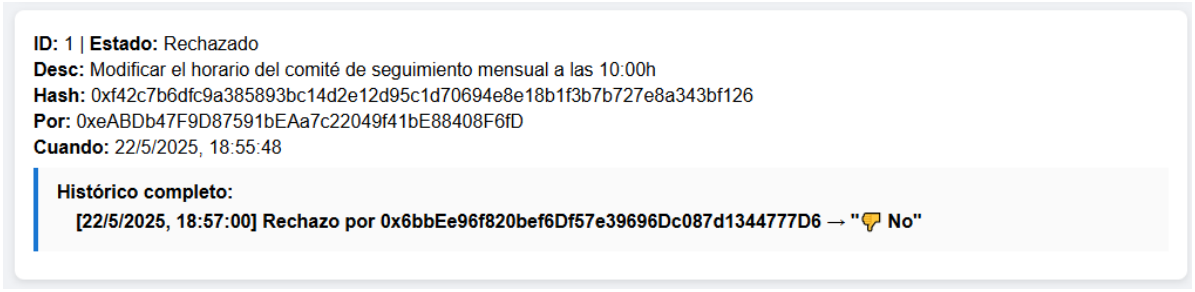

Figura 20. Ejemplo de rechazo de un cambio por parte de un aprobador autorizado [73].

4.3.5. **Cierre de un Cambio**

Una vez que una solicitud de cambio ha sido aprobada por el número mínimo de usuarios requeridos, se habilita la opción de cerrar formalmente dicho cambio. Esta acción está reservada exclusivamente al rol de gestor de proyecto (*Project Manager*), quien es el responsable de confirmar que el procedimiento ha concluido correctamente y que no quedan acciones pendientes.

El cierre de un cambio implica su transición definitiva al estado "Cerrado", lo que marca el final de su ciclo de vida operativo dentro del sistema. Esta operación genera el evento ChangeClosed, que queda registrado en la *blockchain* con el correspondiente sello de tiempo y firma digital, proporcionando garantías de trazabilidad e inmutabilidad.

En la Figura 21, se muestra la interfaz en el momento previo al cierre de un cambio aprobado. El botón "cerrar" aparece habilitado, permitiendo al gestor completar la operación.

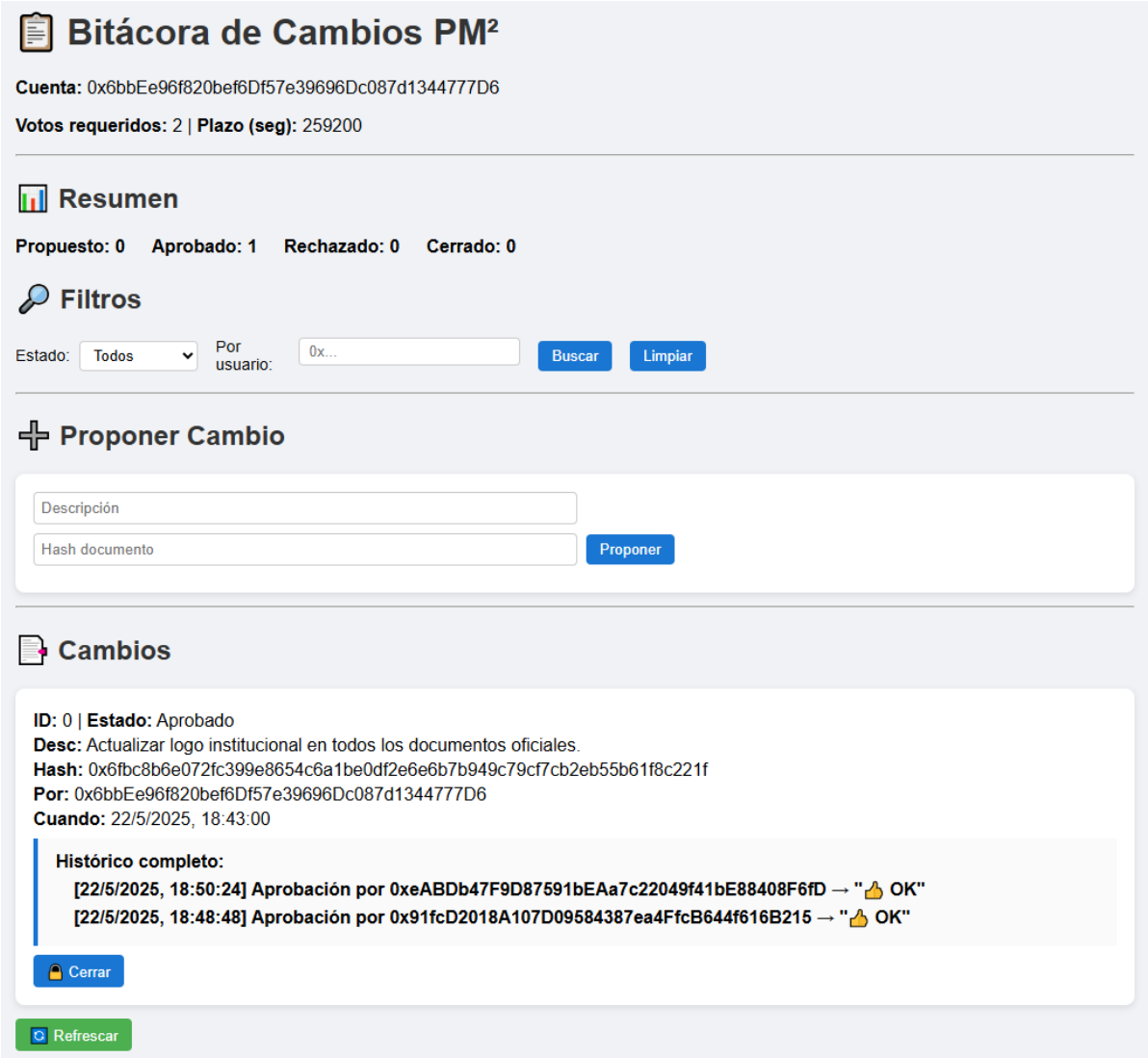

Figura 21. Visualización del estado “aprobado” y opción habilitada para su cierre por parte del gestor de proyecto [73].

4.3.6. **Reapertura de un Cambio**

Cuando un cambio ha sido cerrado, pero se considera necesario reabrirlo (por ejemplo, por nuevos hallazgos o requisitos del proyecto), un aprobador puede invocar la función reopenChange. Esta operación:

- Cambia el estado a 'propuesto'.
- Limpia las aprobaciones previas para permitir una nueva ronda de votaciones.
- Añade una entrada en el historial con motivo de reapertura y fecha.

Es importante destacar que, aunque los votos previos se eliminan funcionalmente, el historial completo de eventos se mantiene (ver Figura 22). Esto garantiza la trazabilidad total del ciclo de vida de la solicitud, incluyendo reaperturas sucesivas.

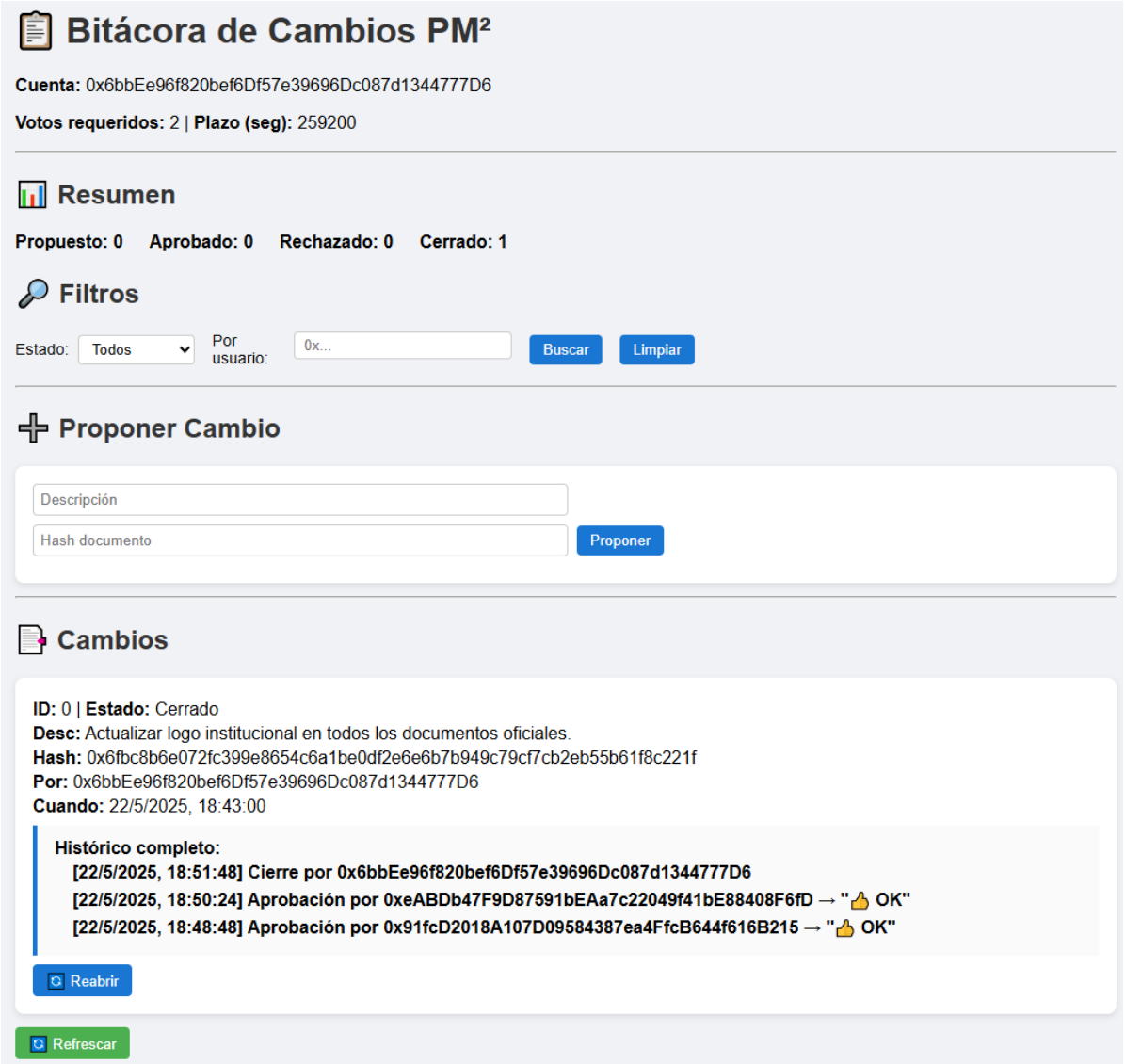

Figura 22. Visualización del cambio ya cerrado,incluyendo el historial completo con trazabilidad de aprobaciones, cierre y evento de reapertura disponible [73].

4.3.7. **Consultas, Filtros y Auditoría**

Una de las funcionalidades clave de la aplicación es la capacidad de consultar y auditar fácilmente los cambios registrados, gracias al uso de filtros dinámicos por estado y por autor (dirección Ethereum). Esto permite a los gestores de proyecto y usuarios validadores obtener rápidamente información relevante según su necesidad.

- **Filtro por estado del cambio:** el usuario puede seleccionar uno de los cuatro estados posibles (propuesto, aprobado, rechazado, cerrado) para visualizar únicamente los cambios que se encuentran en ese estado. Esto simplifica el análisis de qué cambios requieren acción, cuáles están pendientes, y cuáles han finalizado su ciclo. Véase Figura 23.

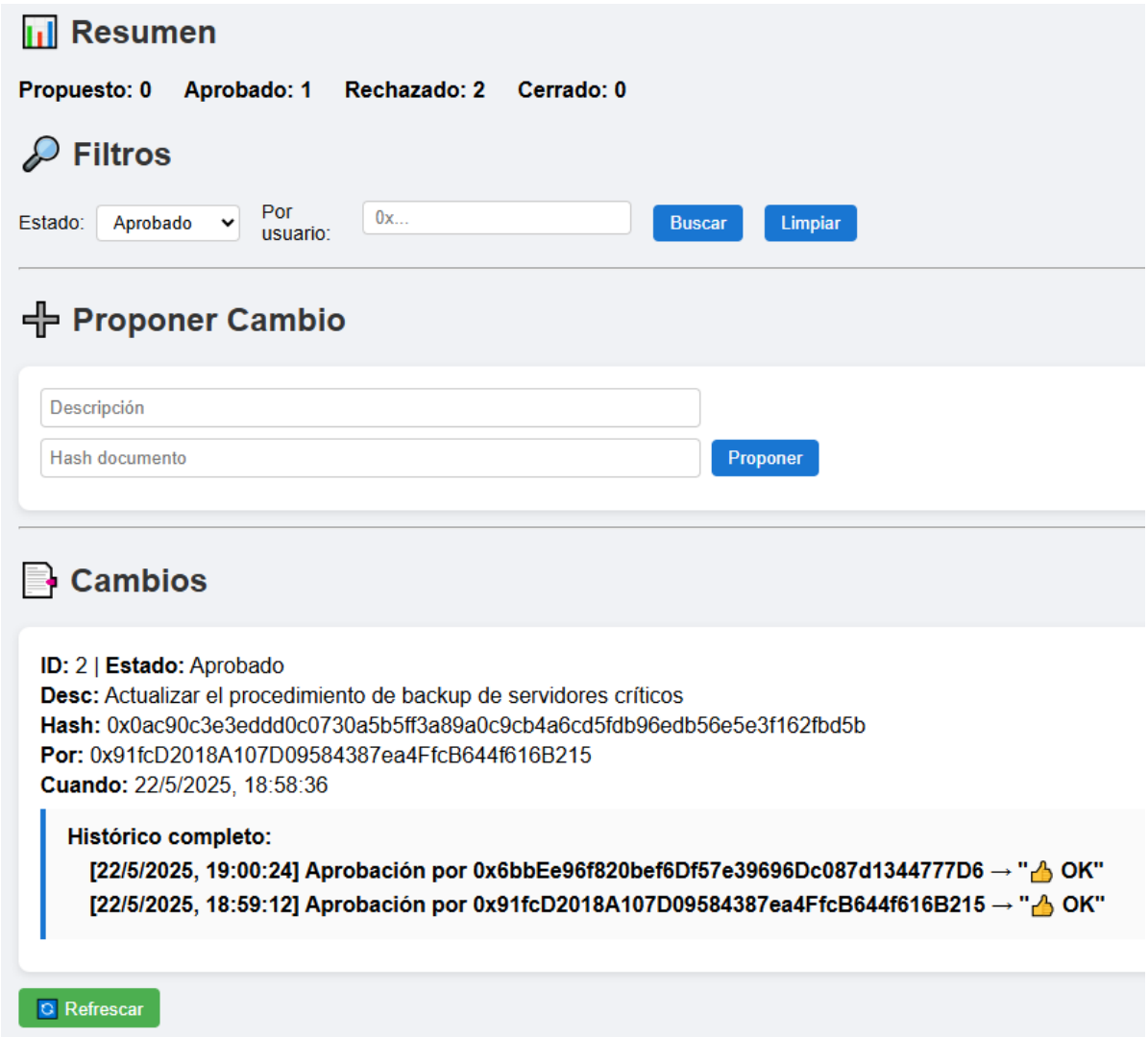

Figura 23.Visualización del uso de filtros por estado del cambio, mostrando únicamente aquellos con estado "aprobado", junto con el resumen de actividad y el historial de votos correspondiente [73].

- **Filtro por dirección del usuario:** también es posible buscar todos los cambios propuestos por una dirección específica. Esta funcionalidad es especialmente útil en contextos colaborativos, donde múltiples usuarios interactúan con la bitácora, y se necesita auditar o revisar los cambios iniciados por un miembro concreto del equipo. Véase Figura 24.

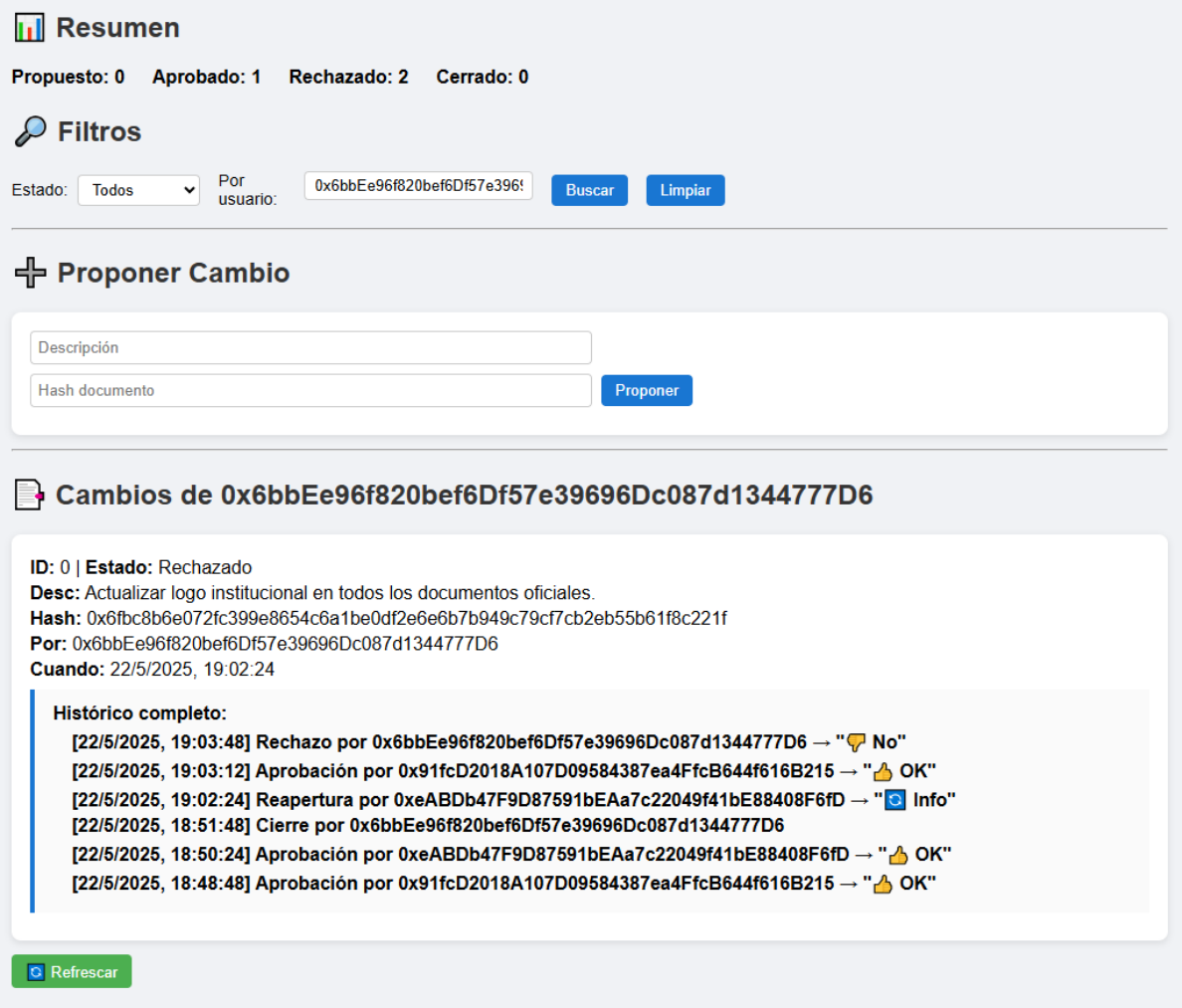

Figura 24. Ejemplo de aplicación del filtro por dirección de usuario, mostrando únicamente los cambios propuestos y gestionados por una cuenta específica en la bitácora [73].

- **Bitácora consolidada:** la interfaz permite ver el histórico completo de cada cambio seleccionado (véase Figura 25), incluyendo:
    - El estado actual del cambio.
    - La secuencia cronológica de eventos (aprobaciones, rechazos, reaperturas, cierres).
    - El autor de cada acción y los comentarios asociados.

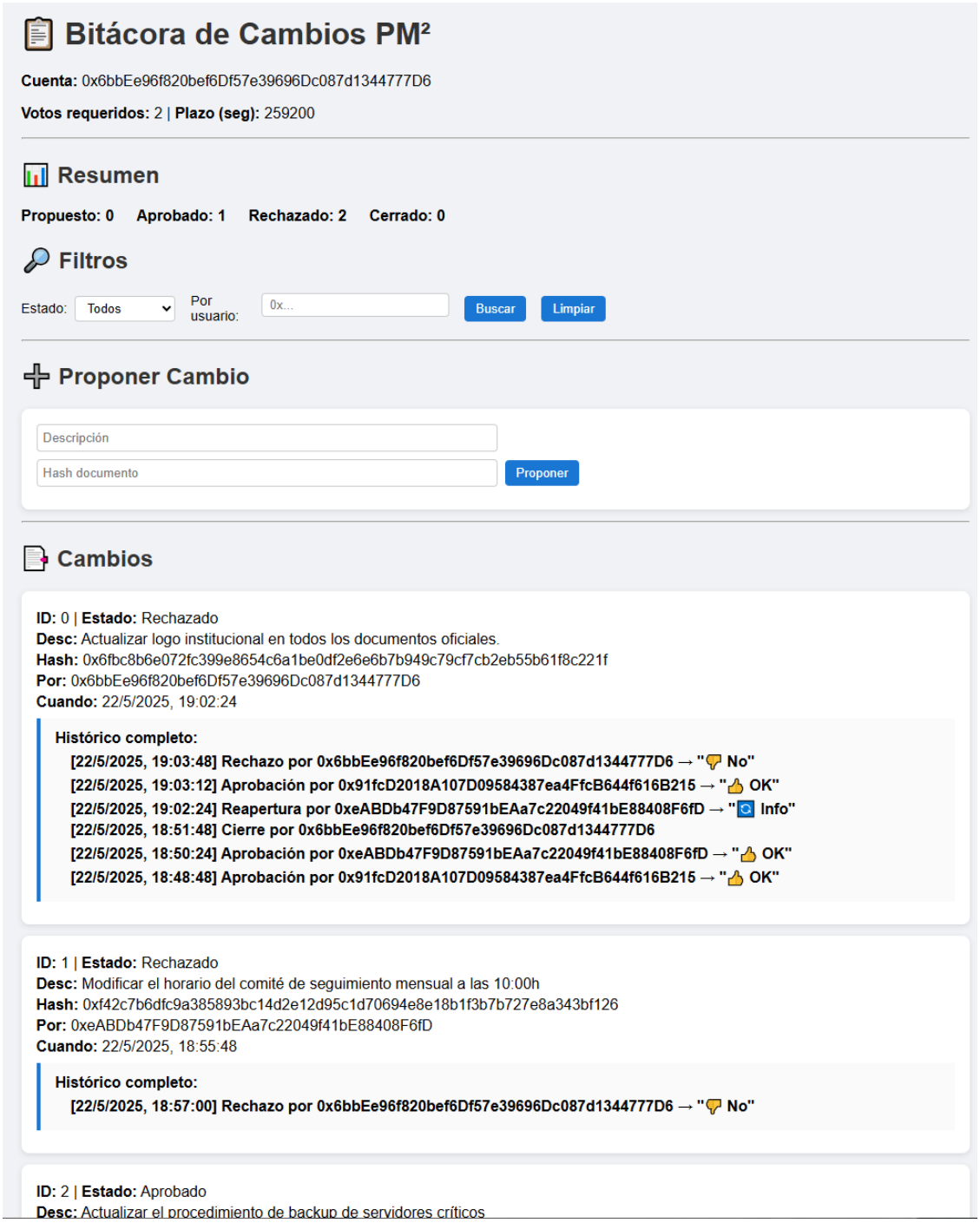

Figura 25. Visualización consolidada de la bitácora de cambios, que muestra el historial completo de eventos asociados a cada solicitud, incluyendo su estado, secuencia de acciones y comentarios emitidos por los usuarios [73].

- **Auditoría externa en Etherscan:** para reforzar la transparencia y la verificabilidad independiente, el sistema se apoya en la trazabilidad *on-chain* de las transacciones. Cada acción ejecutada (propuestas, aprobaciones, rechazos, cierres o reaperturas) queda registrada en la *blockchain* de Ethereum a través del contrato inteligente, de forma segura e inmutable.

  Este historial puede consultarse públicamente en Etherscan [75], simplemente buscando la dirección del contrato (véase Figura 26). Allí se muestran todas las transacciones ejecutadas contra él, incluyendo qué método fue llamado (proposeChange, approveChange, etc.), qué cuenta lo ejecutó, cuándo se realizó la acción (timestamp y bloque), así como el coste de gas y otros metadatos técnicos.

  Esta capacidad de auditoría no requiere acceso a la interfaz gráfica, lo que garantiza la integridad del sistema incluso si el frontend no está disponible.

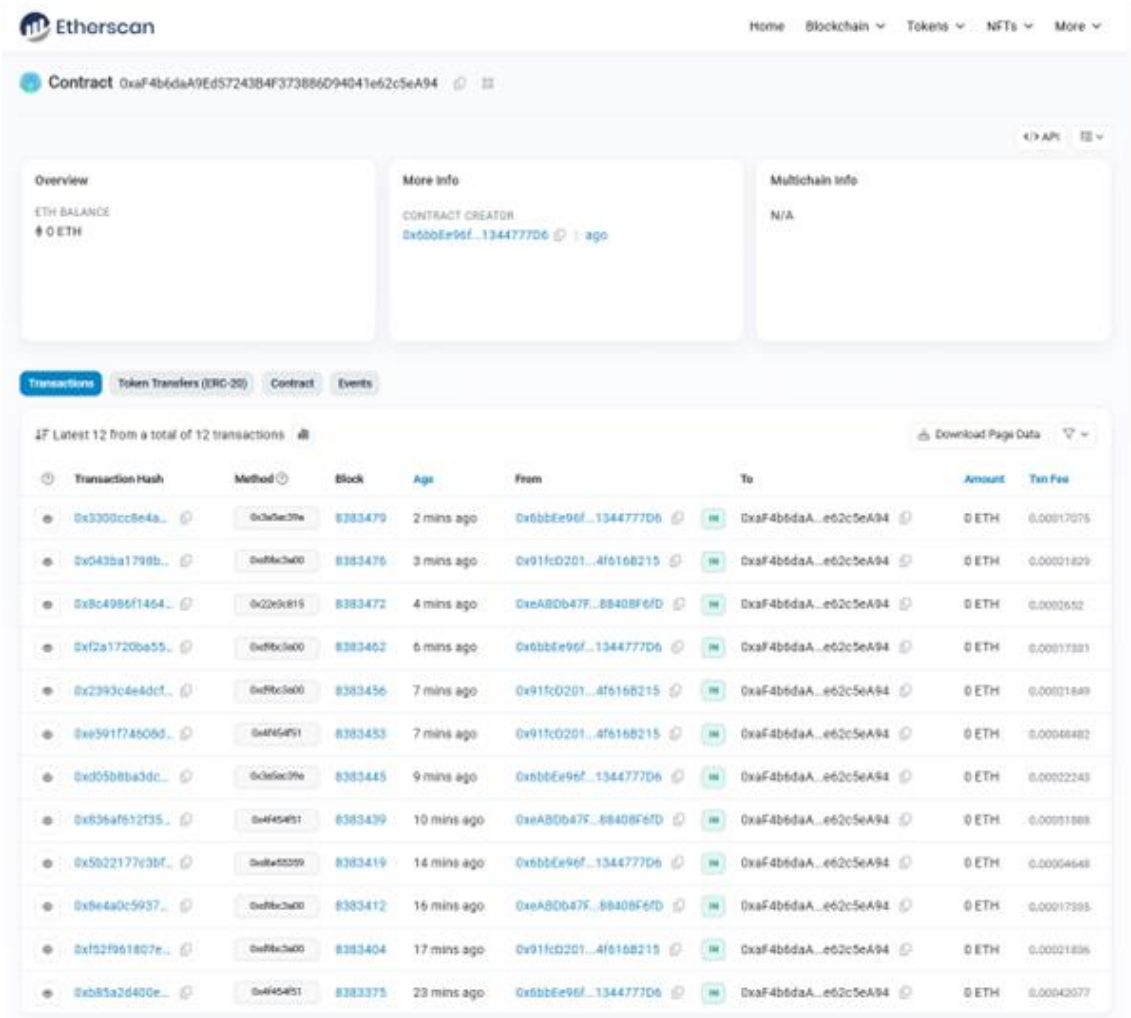

Figura 26. Visualización del contrato inteligente en Etherscan [75], donde se registran todas las transacciones ejecutadas (propuestas, aprobaciones, rechazos, etc.) con sus detalles técnicos, permitiendo auditoría pública y verificación independiente del historial.

4.3.8. **Validación de Errores y Seguridad**

El sistema ha sido probado ante múltiples situaciones no válidas, confirmando que los mecanismos de seguridad y las restricciones del contrato funcionan correctamente. Algunos ejemplos de mensajes de errores pueden ser observados en la Figura 27.

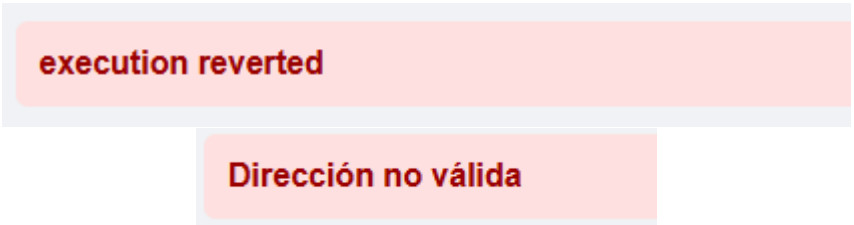

Figura 27. Ejemplo de mensajes de errores mostrados al realizar una petición no válida [73].

4.3.9. **Conclusiones de la Implementación**

La fase de verificación ha permitido confirmar que el sistema desarrollado cumple satisfactoriamente con los requisitos funcionales definidos para una bitácora de cambios alineada con la metodología PM², operando sobre tecnología *blockchain*. Durante las pruebas realizadas se ha verificado que:

- El contrato inteligente puede ser desplegado correctamente en una red pública (Sepolia), y su comportamiento responde con precisión a lo esperado, gracias a un diseño claro de estados, roles y eventos.
- La interfaz web desarrollada permite interactuar de forma intuitiva con la *blockchain*, facilitando al usuario la gestión completa del ciclo de vida de un cambio: propuesta, votación, cierre y reapertura.
- Las funciones de control de acceso y restricciones por rol actúan como salvaguardas efectivas contra acciones indebidas, asegurando que solo los perfiles autorizados puedan ejecutar operaciones críticas.
- El sistema permite trazabilidad total, ya que todos los eventos quedan registrados de forma inmutable en la *blockchain*. Este histórico puede auditarse desde la aplicación o directamente desde Etherscan, sin necesidad de depender de la capa visual.
- Las funcionalidades de filtrado por estado y usuario, así como la reconstrucción cronológica de eventos, ofrecen una herramienta completa tanto para la operación diaria como para la revisión posterior de decisiones.

Además, los escenarios de prueba negativa han demostrado que el sistema responde adecuadamente ante errores de uso (votos duplicados, fuera de plazo, sin permisos), devolviendo mensajes claros al usuario final.

En conjunto, la fase de pruebas ha validado no solo la viabilidad técnica de la solución, sino también su utilidad práctica como herramienta digital para el registro y gestión de cambios en entornos descentralizados. Las capturas incluidas ilustran fielmente esta operatividad y consolidan la fiabilidad del sistema propuesto.

Capítulo 5 Discusión y Conclusiones

Este capítulo ofrece una visión crítica sobre la solución tecnológica propuesta en el presente Trabajo Fin de Máster (TFM), cuyo objetivo ha sido integrar la tecnología *blockchain*, especialmente mediante contratos inteligentes, con los procesos fundamentales de la metodología PM². Tras desarrollar un prototipo funcional centrado en el artefacto Bitácora de Cambios (*Change Log*), se analizan de manera rigurosa tanto los beneficios como las limitaciones técnicas, operativas y organizativas que han surgido durante la implementación y la validación del sistema.

La discusión incluye un análisis económico detallado que compara los costes iniciales de implementación (CAPEX) y los costes operativos recurrentes (OPEX), considerando diversas opciones estratégicas como migraciones hacia redes *blockchain* más económicas (como Polygon) o incluso redes privadas. Además, se realiza una estimación preliminar del retorno de la inversión frente a métodos tradicionales basados en software licenciado.

Asimismo, se evalúa el impacto medioambiental derivado del uso de *blockchain*, destacando los beneficios en eficiencia energética gracias al actual mecanismo de consenso *Proof of Stake* (PoS) de Ethereum y alternativas sostenibles como Polygon, frente al consumo habitual de plataformas convencionales alojadas en servidores físicos.

Finalmente, se proponen diversas líneas de mejora futuras que pueden ampliar significativamente el alcance y la eficacia del sistema presentado, abordando no solo la escalabilidad tecnológica, sino también la integración estratégica con otros artefactos PM², la interoperabilidad documental, el uso eficiente de la documentación derivado de los costes transaccionales asociados, y la inclusión de tecnologías emergentes complementarias como la inteligencia artificial o identidades digitales verificables.

El capítulo concluye sintetizando los hallazgos fundamentales, resaltando la contribución específica del trabajo realizado al campo de la dirección de proyectos bajo PM², y sentando las bases para futuras implementaciones que busquen aprovechar plenamente el potencial disruptivo y operativo de la tecnología *blockchain*.

5.1 Discusión de los Resultados Obtenidos

Los resultados derivados de la implementación del sistema basado en contratos inteligentes revelan que la integración de *blockchain* en la metodología PM² aporta mejoras claras en trazabilidad, transparencia y eficiencia operativa, especialmente en la gestión de artefactos clave como la Bitácora de Cambios (*Change Log*). Este desarrollo práctico no solo valida la teoría previa, sino que también evidencia beneficios concretos en términos de automatización y reducción de cargas administrativas repetitivas.

Una ventaja destacable observada durante las pruebas fue la reducción significativa del esfuerzo manual dedicado por parte del equipo de proyecto en la validación y auditoría de los cambios. La transparencia inherente al registro *blockchain* facilita las auditorías internas y externas, disminuyendo notablemente riesgos de error, manipulación o fraude documental, situaciones frecuentes en métodos convencionales de gestión de información.

No obstante, es fundamental mencionar varias limitaciones y riesgos identificados durante la implementación práctica, los cuales deben ser abordados cuidadosamente antes de una implantación definitiva en un entorno productivo. Estas cuestiones incluyen:

Limitaciones técnicas y operativas

- Dependencia tecnológica y costes variables del gas: la solución implementada actualmente utiliza la red Ethereum *Mainnet*, cuyo coste de transacción (gas) presenta una considerable volatilidad. Aunque la estabilidad tecnológica es robusta, dicha fluctuación puede generar incertidumbre presupuestaria. Una posible solución, ya planteada y que merece un análisis más profundo, sería migrar parcial o totalmente el contrato inteligente a redes alternativas, como Polygon, que ofrecen tarifas significativamente más bajas, o incluso a redes privadas en las que los costes operativos serían prácticamente nulos.

- Complejidad de adopción tecnológica: la tecnología *blockchain* implica una curva inicial de aprendizaje para los usuarios no técnicos. La gestión segura de carteras digitales como MetaMask, la comprensión básica del funcionamiento de las transacciones *on-chain* y la necesidad de proteger adecuadamente las claves privadas constituyen barreras iniciales. Es esencial desarrollar programas formativos continuos para facilitar la transición cultural y tecnológica hacia el entorno *blockchain*.

Riesgos específicos de seguridad y mantenimiento

- Vulnerabilidades potenciales del código: los contratos inteligentes desplegados en *blockchain* son inmutables; por lo tanto, la identificación proactiva de vulnerabilidades mediante auditorías técnicas rigurosas es imprescindible antes del despliegue definitivo. Una vulnerabilidad no detectada podría generar pérdidas económicas o comprometer la integridad del sistema.

- Gestión de claves privadas: el control y la custodia segura de las claves privadas representan un riesgo relevante, especialmente en entornos corporativos o institucionales. Pérdidas o accesos no autorizados podrían generar consecuencias críticas, incluyendo pérdidas económicas o interrupción operativa significativa. La definición de una política robusta de gestión criptográfica es fundamental para mitigar este riesgo.

- Obsolescencia tecnológica: el rápido avance tecnológico en el ámbito *blockchain* requiere anticipar posibles actualizaciones o migraciones a nuevas redes o versiones del software. Una planificación tecnológica flexible y evolutiva puede reducir significativamente el impacto futuro de la obsolescencia.

En términos económicos, la comparación inicial sugiere un ahorro considerable al adoptar *blockchain* frente a soluciones basadas en licencias de software tradicionales. Mientras que estas licencias suelen implicar costes fijos altos, la *blockchain* ofrece un modelo de costes vinculados al uso efectivo, lo que fomenta una gestión documental más eficiente. Este modelo de pago por transacción actúa indirectamente como incentivo hacia el uso racionalizado y estratégico de la documentación generada en los proyectos.

Asimismo, la posible migración a redes alternativas, especialmente Polygon o redes privadas, reduciría aún más los costes operativos recurrentes, mejorando sustancialmente el atractivo financiero a medio y largo plazo. Esta flexibilidad económica y tecnológica permite un enfoque más adaptado a cada escenario organizacional concreto, fortaleciendo la propuesta frente a métodos convencionales que implican rigideces presupuestarias significativas.

En síntesis, los resultados obtenidos demuestran la viabilidad y ventajas prácticas de integrar *blockchain* con PM², al tiempo que señalan claramente las limitaciones y riesgos que deben gestionarse estratégicamente. El reconocimiento temprano de estos elementos críticos contribuye decisivamente a

garantizar la sostenibilidad y efectividad del sistema propuesto en contextos reales, aportando valor tangible y estratégico a las organizaciones que opten por su adopción.

5.2 Estudio Económico

La viabilidad económica es una dimensión crítica que debe analizarse cuidadosamente antes de considerar cualquier implementación tecnológica. En este apartado se presenta un análisis económico preliminar de la solución propuesta basada en contratos inteligentes desplegados sobre Ethereum, considerando tanto los costes iniciales de implantación (CAPEX) como los costes recurrentes operativos (OPEX). Las cifras mostradas a continuación han sido estimadas considerando precios y condiciones reales vigentes a junio de 2025, con el fin de presentar un escenario lo más fiel posible a la realidad del mercado.

5.2.1. **Costes de Implantación Inicial (CAPEX)**

Los costes iniciales incluyen todos aquellos gastos relacionados con la puesta en marcha y configuración inicial del sistema *blockchain* propuesto. Su desglose se muestra en la Tabla 11.

Tabla 11. Desglose de los gastos de capital (CAPEX). Elabroación propia con Microsoft Excel [71].

Concepto	Descripción	Coste estimado (€)
Diseño y desarrollo del contrato inteligente (Solidity)	Programación y configuración inicial del contrato inteligente	8.000
Auditoría técnica externa del contrato inteligente	Revisión y validación por auditores externos expertos para prevenir vulnerabilidades	6.000
Desarrollo del *frontend* (interfaz web React)	Diseño e implementación de interfaz gráfica para usuarios no técnicos	6.500
Integración y pruebas técnicas	Pruebas exhaustivas del contrato y la aplicación, garantizando su integración y correcto funcionamiento	3.500
Despliegue inicial en red Ethereum (*Mainnet*)	Coste inicial de gas para desplegar el contrato inteligente en la red principal Ethereum	1.500

Concepto	Descripción	Coste estimado (€)
Formación inicial del equipo usuario	Sesiones formativas iniciales para usuarios del sistema sobre *blockchain* y uso práctico del sistema	2.000
Total CAPEX	**Inversión total inicial necesaria para la puesta en marcha del proyecto**	**27.500 €**

El total del CAPEX asciende a VEINTISIETE MIL QUINIENTOS EUROS (27.500 €).

Estos costes iniciales reflejan la inversión necesaria para garantizar una puesta en producción segura y efectiva. En particular, cabe destacar que la auditoría externa del contrato inteligente es esencial para mitigar posibles vulnerabilidades y errores críticos, por lo que constituye un gasto significativo, pero absolutamente necesario en cualquier entorno empresarial o institucional.

5.2.2. **Costes Operativos Recurrentes (OPEX)**

Los costes operativos se relacionan con los gastos recurrentes en los que incurre la organización durante la operación normal del sistema. Estos incluyen principalmente las tarifas de transacción en Ethereum, servicios en la nube auxiliares y costes de mantenimiento periódico. Su desglose se muestra en la Tabla 12.

Tabla 12. Desglose de los gastos operativos (OPEX) anuales. Elaboración propia con Microsoft Excel [71].

Concepto	Descripción	Coste anual estimado (€)
Costes de gas en Ethereum (transacciones en *Mainnet*)	Asumiendo un promedio de 60 transacciones mensuales (720 anuales) con coste medio aproximado de 5€ por transacción (según precios vigentes en junio de 2025).	3.600
Almacenamiento descentralizado (IPFS/Filecoin)	Almacenamiento seguro y descentralizado para documentos justificativos, estimado en aproximadamente 1 TB anuales con coste mensual de 100€.	1.200

Concepto	Descripción	Coste anual estimado (€)
Mantenimiento y soporte técnico del sistema	Revisiones trimestrales de seguridad, mejoras evolutivas periódicas, soporte permanente y resolución de incidencias por parte de un proveedor externo especializado.	4.500
Licencias y herramientas auxiliares	Suscripciones para herramientas auxiliares como monitorización *blockchain*, alertas de seguridad, análisis de rendimiento y generación automática de reportes.	800
Formación continua y soporte al usuario	Formación periódica adicional y soporte continuo a usuarios del sistema.	1.000
Total OPEX anual	**Coste anual recurrente estimado para operar el sistema en producción**	**11.100 €**

El total del OPEX anual asciende a ONCE MIL CIEN EUROS (11.100 €).

5.2.3. **Posible Migración a Otras Redes *Blockchain* para Reducción de Costes**

Una estrategia clave para reducir significativamente los costes operativos (OPEX) identificados en la implementación sobre Ethereum *Mainnet* es la posible migración o extensión de la solución hacia otras redes *blockchain* que ofrezcan menores costes de transacción o una estructura tarifaria más predecible.

- **Red Pública Polygon (Layer-2)**: Polygon es una red *blockchain* pública, clasificada como solución de capa 2, compatible con *Ethereum Virtual Machine* (EVM), lo cual simplifica notablemente el proceso de migración desde Ethereum *Mainnet*. En junio de 2025, las transacciones en Polygon tienen un coste promedio inferior a 0,01 € por transacción, significativamente más bajo que los aproximadamente 5 € en Ethereum [28,35].

  Suponiendo el mismo volumen de transacciones anual previsto para el piloto realizado (720 transacciones al año), el coste anual en Polygon sería inferior a 100 €, frente a los 3.600 € de Ethereum. El coste de almacenamiento descentralizado (IPFS/Filecoin) y soporte técnico se mantendría estable en 7.500 €, por lo que el OPEX anual total se reduciría hasta aproximadamente 7.600 €.

  El CAPEX asociado a migrar el contrato inteligente actual a Polygon, considerando pruebas de integración, auditoría adicional específica para la red y formación técnica al equipo, se estima en aproximadamente 2.500 € adicionales, dada la compatibilidad con EVM.

- **Redes Privadas (Hyperledger Fabric o Quorum)**: las redes *blockchain* privadas ofrecen control total sobre infraestructura y costes operativos. Redes como Hyperledger Fabric o Quorum eliminan completamente los costes variables por transacción, pero requieren una inversión inicial superior para desplegar la infraestructura dedicada (servidores, nodos, configuración específica).

  La implantación inicial en redes privadas para el piloto se estima en torno a 15.000 € adicionales, cubriendo infraestructura, configuración y auditoría específica [38]. En términos de OPEX, la eliminación de costes variables permite reducir notablemente el coste recurrente anual, que se situaría alrededor de los 5.900 €, considerando mantenimiento, soporte técnico y almacenamiento descentralizado.

  La decisión sobre una red privada debe considerar adicionalmente la necesidad de privacidad, confidencialidad y control absoluto sobre los datos, ventajas que justifican el coste adicional inicial en determinados entornos altamente regulados.

5.2.4. Estimación del Coste Total para Implementar *Blockchain* en todos los Artefactos PM² (33 Artefactos)

Este trabajo ha alcanzado un nivel TRL 6, correspondiente a un prototipo funcional demostrado en entorno relevante [80], lo que permite hacer estimaciones razonables sobre la inversión total requerida para implementar *blockchain* en el conjunto completo de artefactos definidos por la metodología PM².

Considerando la agrupación lógica y funcional de los artefactos PM², es posible estimar que podrían ser cubiertos eficazmente mediante el desarrollo de aproximadamente 10 módulos *blockchain* diferentes (por ejemplo, módulos específicos para *Change Log*, *Change Request Form*, *Risk Log*, *Deliverable Acceptance Form*, entre otros similares), compartiendo interfaces comunes y lógicas reutilizables.

Los costes estimados de una implementación completa serían los siguientes (considerando como base los costes del prototipo desarrollado):

- Diseño y desarrollo *blockchain* (10 módulos): 10 módulos × 8.000 € cada uno = 80.000 €.
- Auditoría técnica *blockchain* (10 módulos): 10 módulos × 6.000 € cada uno = 60.000 €.
- Desarrollo del *frontend* (adaptaciones adicionales): Estimado global adicional = 25.000 €.
- Integración, pruebas técnicas y formación ampliada: Costes incrementales por escala adicional = 20.000 €.

Por lo tanto, el CAPEX estimado total para implementar *blockchain* en los 33 artefactos PM² se aproxima a los 185.000 €.

Respecto a los costes operativos anuales estimados para una implantación completa:

- Ethereum *Mainnet*: estimando aproximadamente 4.800 transacciones anuales (400 mensuales debido a mayor cobertura de artefactos), el coste anual solo en transacciones sería de unos 24.000 €, elevando el OPEX anual a aproximadamente 31.500 €.

- Polygon: con un coste de transacción mucho menor, el OPEX anual total podría reducirse considerablemente hasta cerca de 8.000 €.

- Red privada: eliminando costes variables, el OPEX anual total se reduciría hasta unos 5.900 €.

5.2.5. **Análisis Global y Retorno de Inversión Estimado (ROI)**

Para evaluar la viabilidad económica real del proyecto, resulta fundamental realizar un análisis detallado del retorno de la inversión inicial (ROI). Este análisis se presenta a continuación, diferenciando claramente dos escenarios:

- Escenario 1: implementación específica de *blockchain* para la gestión del artefacto *Change Log* (realizada en este TFM).

- Escenario 2: implementación completa de *blockchain* en el conjunto de los 33 artefactos definidos en la metodología PM².

Escenario 1: ROI del proyecto actual (artefacto *Change Log*)

La implementación desarrollada en este trabajo presenta una inversión inicial (CAPEX) estimada de 27.500 € con costes operativos anuales (OPEX) sobre Ethereum *Mainnet* de 11.100 €. Comparativamente, los costes anuales típicos asociados a la gestión tradicional del artefacto *Change Log* en organizaciones medianas están estimados en un rango entre 20.000 y 25.000 €, incluyendo costes administrativos, licencias de software específicas y tareas manuales repetitivas [81].

Considerando un ahorro anual conservador de 13.000 € en tareas manuales, reducción de licencias software específicas y eficiencia administrativa, el retorno de la inversión inicial para el prototipo específico del *Change Log* es el siguiente:

- Ethereum *Mainnet* (CAPEX: 27.500 €; OPEX: 11.100 €; ahorro anual: 13.000 €):
    - ROI estimado ≈ 14 años (considerando la alta tarifa por transacción).
- Polygon (CAPEX adicional: 2.500 €, total 30.000 €; OPEX anual reducido a 7.600 €):
    - ROI estimado ≈ 6 años, significativamente más atractivo debido al menor coste de transacción.
- Red Privada (CAPEX adicional: 15.000 €, total 42.500 €; OPEX anual 5.900 €):
    - ROI estimado ≈ 6 años, equilibrado con las ventajas adicionales en seguridad y privacidad.

Es evidente que, debido a los elevados costes actuales de Ethereum *Mainnet*, resulta especialmente recomendable migrar la solución hacia Polygon o redes privadas para obtener un retorno de inversión aceptable en un periodo razonable.

Escenario 2: ROI estimado de la implementación *blockchain* completa (33 artefactos PM²)

Para la implantación completa del modelo *blockchain* cubriendo los 33 artefactos PM², la inversión inicial total estimada es considerablemente mayor, alcanzando los 185.000 € sobre Ethereum, 187.500 € sobre Polygon y 200.000 € en redes privadas (véase sección anterior 5.2.4).

Por otro lado, la gestión documental y operativa tradicional, utilizando herramientas comerciales estándar como ERP o software especializado de gestión de proyectos, genera costes anuales totales significativamente mayores, en torno a 40.000-80.000 € al año, según estimaciones del mercado para organizaciones medianas [81].

Asumiendo una media conservadora de 60.000 € de costes operativos anuales en gestión tradicional, y un ahorro anual realista entre 20.000 € y 30.000 € mediante automatización, reducción de licencias software específicas y menor carga administrativa, los retornos de inversión serían los siguientes:

- Ethereum *Mainnet* (CAPEX: 185.000 €; OPEX: 31.500 €; ahorro anual medio: 25.000 €):
    - Ahorro neto anual: 25.000 €.
    - ROI estimado ≈ 7 años y medio.
- Polygon (CAPEX: 187.500 €; OPEX: 8.000 €; ahorro anual medio: 25.000 €):
    - Ahorro neto anual: 52.000 € (60.000 tradicional – 8.000 *blockchain*).
    - ROI estimado ≈ 3 años y medio, muy atractivo y rápidamente recuperable.
- Red Privada (CAPEX: 200.000 €; OPEX: 5.900 €; ahorro anual medio: 25.000 €):
    - Ahorro neto anual: 54.100 € (60.000 tradicional – 5.900 *blockchain*).
    - ROI estimado ≈ 3 años y 8 meses, también favorable, especialmente considerando la privacidad y seguridad adicionales.

Comparando ambos escenarios, queda claro que, aunque la implementación completa de *blockchain* para todos los artefactos PM² representa una inversión inicial significativamente superior, su potencial de ahorro a largo plazo es también mayor. Especialmente atractiva resulta la migración hacia redes *blockchain* con costes más bajos, como Polygon o redes privadas, dado que el ROI se recupera en menos de cuatro años.

Por tanto, desde un punto de vista estrictamente económico, la opción de escalar el piloto actual a todos los artefactos de PM² es altamente recomendable, siempre que se opte por una red *blockchain* económicamente viable (Polygon o privada), garantizando una inversión sostenible, rápida recuperación del capital invertido y beneficios significativos en términos de eficiencia operativa, seguridad documental y reducción sustancial de costes administrativos recurrentes.

5.2.6. **Consideraciones Económicas Finales**

La elección definitiva de la red *blockchain* dependerá de factores específicos del entorno en el que se implemente, incluyendo requisitos regulatorios, grado de confidencialidad de datos, y sensibilidad al coste variable de las transacciones.

Considerando la estimación económica realizada, resulta especialmente atractivo evaluar una migración parcial o total hacia Polygon para maximizar la eficiencia económica a largo plazo, especialmente en contextos menos críticos desde el punto de vista regulatorio.

Alternativamente, en escenarios de elevada criticidad o sensibilidad de datos (por ejemplo, organismos gubernamentales, sectores altamente regulados como el financiero o la salud), la inversión adicional en una red privada puede justificarse plenamente por la seguridad y confidencialidad superior ofrecida. En definitiva, disponer de estas alternativas estratégicas dota al proyecto de una flexibilidad clave para adaptarse a diferentes escenarios y requisitos operativos, garantizando la sostenibilidad económica y técnica de la solución a largo plazo.

5.3 Análisis de Impacto Medioambiental

En un contexto de creciente conciencia ambiental, toda solución tecnológica debe ser evaluada no sólo en términos de eficiencia y coste, sino también en función de su huella ecológica. En este apartado se analiza el posible impacto medioambiental derivado del uso de tecnología *blockchain* para la gestión de artefactos PM², con especial atención al consumo energético asociado a las operaciones *on-chain*.

5.3.1. **Consumo Energético de la *Blockchain***

Históricamente, las redes *blockchain* han sido objeto de críticas por su elevado consumo energético, especialmente aquellas que utilizaban el mecanismo de consenso *Proof of Work* (PoW), como era el caso de Ethereum antes de su transición. Sin embargo, desde septiembre de 2022, Ethereum opera bajo el mecanismo *Proof of Stake* (PoS), lo que ha supuesto una reducción del consumo energético superior al 99,9 %, según estimaciones de la Ethereum *Foundation*.

Actualmente, una única transacción en Ethereum bajo PoS consume menos energía que una búsqueda en Google o una operación de tarjeta bancaria estándar. Por tanto, el uso de Ethereum en su estado actual presenta un impacto ambiental muy reducido, especialmente si se compara con sistemas de gestión documental tradicionales alojados en centros de datos intensivos.

Además, soluciones alternativas como Polygon, mencionada en el apartado anterior por su eficiencia de costes, son aún más sostenibles desde un punto de vista energético. Polygon no solo utiliza PoS, sino que ha invertido en programas de compensación de carbono, posicionándose como una red climáticamente neutra desde 2022. Esto convierte a Polygon en una alternativa especialmente recomendable desde el punto de vista ambiental.

5.3.2. **Comparativa frente a Soluciones Tradicionales**

Las plataformas de gestión de proyectos convencionales, que suelen depender de servidores centralizados, presentan un impacto ambiental que no siempre es evidente: consumo energético

constante en centros de datos, refrigeración, mantenimiento de hardware, y redundancia geográfica para evitar pérdidas de información. Aunque el modelo SaaS ha permitido cierto grado de optimización energética, no elimina la dependencia de infraestructuras físicas de alto consumo.

Por el contrario, la solución propuesta en este trabajo (basada en *smart contracts* y almacenamiento descentralizado) reduce la necesidad de servidores dedicados y externaliza parte del procesamiento a una red distribuida que ya está activa, compartida entre miles de nodos que operan con una eficiencia energética cada vez mayor.

Oportunidades de mejora ambiental

A pesar del bajo impacto asociado, existen estrategias complementarias que permitirían maximizar la sostenibilidad ambiental del sistema:

- Migrar a redes climáticamente neutras como Polygon, Optimism o Arbitrum, siempre que mantengan compatibilidad EVM y niveles aceptables de descentralización.
- Reducir la cantidad de datos almacenados directamente en la *blockchain*, favoreciendo referencias mediante *hashes* y el uso de IPFS para ficheros pesados.
- Minimizar transacciones innecesarias mediante una lógica eficiente de eventos y uso selectivo de funciones *view* en lugar de write, lo cual reduce tanto el coste como el consumo energético.
- Integrar prácticas de *green coding* en la aplicación *frontend*, optimizando el uso de recursos del navegador y el renderizado reactivo.

5.3.3. Conclusión Ambiental

En breve síntesis, la solución propuesta se encuentra alineada con una visión responsable del desarrollo tecnológico, gracias a la elección de infraestructuras *blockchain* modernas, sostenibles y altamente eficientes desde el punto de vista energético. La migración a redes climáticamente neutras, junto con prácticas de desarrollo consciente, puede incluso situar esta solución como una opción más ecológica que muchas plataformas de gestión documental convencionales. Esto fortalece su adecuación no solo funcional y económica, sino también ética y ambiental.

5.4 Futuras Líneas de Mejora

Aunque la solución presentada en este trabajo representa un prototipo funcional y operativo, el potencial de los contratos inteligentes en la gestión documental basada en la metodología PM² va mucho más allá. A continuación, se exponen varias líneas de mejora que permitirían escalar, enriquecer o adaptar el sistema a contextos reales de mayor complejidad, madurez o exigencia.

1. Expansión a múltiples artefactos PM² con módulos interoperables

Actualmente, el sistema se centra en la Bitácora de Cambios (*Change Log*). Una mejora lógica sería extender el contrato inteligente con módulos adicionales para cubrir otros artefactos de alto impacto como:

- *Change Request Form* (CRF).
- *Deliverables Acceptance Plan* (DAP).

- *Risk Log* o *Issues Log* (para trazabilidad de incidentes).

Estos módulos podrían interoperar entre sí mediante eventos cruzados o incluso una línea base de producto (*product baseline*) *on-chain*, creando un ecosistema documental distribuido, con lógica compartida y validación recíproca.

2. Integración con identidades digitales verificables (DID)

Una mejora crítica sería vincular las acciones de los usuarios con identidades digitales verificables (DID), conforme a los estándares del W3C. Esto permitiría reemplazar direcciones públicas anónimas por entidades identificadas (por ejemplo, usuarios certificados, empleados, proveedores), manteniendo trazabilidad sin comprometer la privacidad.

Esto facilitaría también la integración con sistemas de autenticación existentes (LDAP, OAuth, etc.), haciendo viable la adopción en entornos corporativos sin necesidad de reinventar los procesos de acceso.

3. Gobernanza dinámica y personalizable por proyecto

En su versión actual, el contrato asume reglas fijas (por ejemplo, número de votos requeridos o duración del plazo de aprobación). Una línea de mejora sería permitir una gobernanza programable por proyecto, donde ciertos parámetros puedan definirse al inicio de cada iniciativa y adaptarse a la naturaleza o criticidad del proyecto.

Esto permitiría, por ejemplo, establecer mayorías cualificadas para proyectos de alto impacto, reducir el plazo de evaluación en entornos ágiles, o exigir aprobación por roles jerárquicos específicos.

4. Auditoría inteligente basada en IA y alertas proactivas

Podría integrarse una capa de análisis inteligente que, mediante técnicas de *machine learning* o reglas predefinidas, detecte patrones anómalos en los registros de cambios:

- Cambios reiteradamente rechazados por el mismo revisor.
- Aprobaciones emitidas en tiempo récord (potencial riesgo de automatismo indebido)
- Cambios reabiertos repetidamente sin cierre definitivo

Esto permitiría generar alertas tempranas para auditores o PMO, identificando posibles focos de riesgo o incumplimientos de gobernanza antes de que se materialicen.

5. Optimización de costes mediante redes híbridas y multicapa

Otra mejora factible sería rediseñar el sistema para operar de forma híbrida entre diferentes capas de *blockchain*:

- Operaciones críticas y de validación final en redes como Ethereum o Polygon.
- Operaciones de consulta o carga masiva de datos en soluciones L2 como Arbitrum o zkSync.
- Persistencia de documentos pesados en IPFS o Filecoin, con anclaje por *hash*.

Esto permitiría reducir significativamente el OPEX y mejorar la escalabilidad, sin sacrificar seguridad ni interoperabilidad.

6. Interfaces móviles y accesibilidad multiplataforma

Aunque la aplicación actual está orientada al entorno de escritorio, una evolución natural sería el diseño de una interfaz responsive o aplicación móvil (por ejemplo, usando React Native o Flutter), que permitiera a los participantes:

- Proponer y aprobar cambios desde el móvil.
- Recibir notificaciones *push* sobre tareas pendientes.
- Consultar el estado de su proyecto en tiempo real desde cualquier lugar.

Además, se podrían incluir funcionalidades accesibles para usuarios con discapacidad (lectores de pantalla, navegación por teclado, etc.), alineándose con los principios de inclusión digital.

7. Certificación automática de entregables y auditorías

Una evolución interesante sería permitir que, una vez aprobado un cambio o entregable, el sistema emitiera de forma automática un certificado digital firmable (PDF con firma electrónica + hash de validación), que sirva como prueba oficial de aceptación o conformidad.

Esto facilitaría su presentación en auditorías internas o externas, cumpliendo requisitos normativos como ISO 9001, CMMI o incluso auditorías contractuales o gubernamentales.

Estas líneas de mejora no sólo ampliarían las funcionalidades del sistema, sino que lo acercarían progresivamente a una plataforma de gestión documental certificada, modular y auditable, perfectamente alineada con los principios de PM² y con los requisitos operativos del mundo real. Son también una invitación a futuras investigaciones o desarrollos empresariales que deseen llevar esta visión desde el entorno académico a la práctica profesional.

5.5 Conclusiones del Trabajo

Este trabajo ha tenido como objetivo explorar, diseñar e implementar una solución técnica que combine los principios de la metodología PM² con el potencial de la tecnología *blockchain*, a través del uso de contratos inteligentes desplegados en una red pública descentralizada.

Partiendo de un análisis riguroso de los 33 artefactos de PM², se identificaron aquellos con mayor valor potencial al ser digitalizados de forma inmutable, verificable y autoejecutable. Cinco artefactos (el *Project Charter*, el *Change Log*, el *Change Request Form*, el *Deliverable Acceptance Form* y el *Deliverables Acceptance Plan*) fueron seleccionados como casos de uso prioritarios para su implantación *on-chain*.

Sobre esta base, se desarrolló un sistema funcional que incluye un contrato inteligente en Solidity para la gestión de cambios, y una aplicación *frontend* React que permite a los usuarios interactuar con la *blockchain* de forma segura y accesible. La solución fue desplegada con éxito en la red de pruebas Sepolia de Ethereum, y se validó su operatividad a través de múltiples pruebas, tanto funcionales como de seguridad, control de acceso y trazabilidad.
Los resultados obtenidos demuestran que:

- La tecnología *blockchain* es plenamente aplicable a escenarios reales de gestión de proyectos, especialmente en lo relativo a trazabilidad, control de cambios, gobernanza y responsabilidad compartida.

- Es posible reemplazar registros manuales o centralizados por artefactos PM² autoejecutables, cuya lógica y estado quedan anclados de forma inmutable y pública.

- La experiencia de usuario no tiene por qué ser sacrificada en favor de la seguridad: mediante una interfaz intuitiva, se logra un acceso sencillo a un sistema complejo en su arquitectura subyacente.

- Los costes de implementación, aunque no despreciables, son asumibles y escalables, y pueden reducirse significativamente mediante redes alternativas como Polygon.

Desde una perspectiva metodológica, el trabajo contribuye a abrir una línea concreta de innovación en el ámbito de la gestión de proyectos: la convergencia entre buenas prácticas estructuradas (como PM²) y tecnologías descentralizadas que permiten transparencia, resistencia a la manipulación y validación automática de procesos.

A pesar de ello, también se han identificado limitaciones y retos relevantes, como la necesidad de una gobernanza clara sobre las claves criptográficas, la formación de los usuarios en entornos Web3, y la necesidad de infraestructuras interoperables que no dependan exclusivamente del entorno *blockchain*. Las líneas de mejora propuestas (como la extensión a otros artefactos, el uso de identidades verificables, la gobernanza programable o la auditoría automatizada) ofrecen un camino de evolución prometedor para quienes deseen adoptar esta visión en contextos reales.

Como cierre, se debe señalar que este estudio no pretende reemplazar los marcos existentes de gestión de proyectos, sino potenciar su ejecución con herramientas modernas que refuercen los principios que ya defienden: transparencia, trazabilidad, participación y mejora continua. En ese sentido, *blockchain* no es el fin, sino el medio: una infraestructura que, bien aplicada, puede ayudar a cerrar la brecha entre lo que un proyecto documenta y lo que realmente ocurre.

En definitiva, este trabajo fin de máster demuestra que es posible integrar de forma armónica dos mundos a menudo percibidos como opuestos: la rigurosidad metodológica y la innovación tecnológica. Y al hacerlo, abre la puerta a una nueva generación de soluciones de gestión de proyectos (más seguras, más auditables, más justas) para los desafíos organizativos del presente y del futuro.

BIBLIOGRAFÍA

[1] M. Sobrino, J. Pajares, J. Adiego, F. Acebes, Speeding up negotiations for prototypes in an industrial project thanks to blockchain, in: The 19th International Conference on Industrial Engineering and Industrial Management (ICIEM), 2023.

[2] M. Sobrino, J. Pajares, N. Martín-Cruz, F. Acebes, Blockchain to improve prototypes management in a project, in: 29th International Congress on Project Management and Engineering, 2023.

[3] J. Adiego, N. Martín-Cruz, Enhancing Supply Chain Security through Experiential Learning in a Blockchain-Based Decentralized Simulation Platform, in: Educational Solution Category of the EFLE Annual Conference, 2025.

[4] Experiential Learning tools to obtain SMART supply chain competences. Money flow: blockchain. Project Reference: 2022-1-FI01-KA220-HED-000086152, 2022.

[5] Project Management Institute (PMI), https://www.pmi.org/, (2025).

[6] H. Kerzner, Project Management: A Systems Approach to Planning, Scheduling, and Controlling, 2017.

[7] https://ipma.world/ipma-standards-development-programme/icb4/, (2025).

[8] A. Limbach, Managing Successful Projects with PRINCE2 ®, 2017.

[9] https://agilemanifesto.org/, (2025).

[10] H. Kniberg, M. Skarin, Kanban and Scrum: Making the Most of Both, 2010. http://www.lulu.com/content/7731694.

[11] Project Management Methodology Guide 3.0, (n.d.). https://doi.org/10.2799/755246.

[12] H. Kerzner, Al Zeitoun, R. Viana, Project Management Next Generation: The Pillars for Organizational Excellence, 2022.

[13] D. Tapscott, A. Tapscott, Blockchain Revolution: How the Technology Behind Bitcoin and Other Cryptocurrencies Is Changing the World, 2nd ed., 2018.

[14] K. Schwab, The Fourth Industrial Revolution, 2016. www.weforum.org.

[15] N. Szabo, Formalizing and Securing Relationships on Public Networks, First Monday 2 (1997). https://doi.org/10.5210/fm.v2i9.548.

[16] https://ethereum.org/en/whitepaper/, (2025).

[17] Y. Liu, B. Zhao, Z. Zhao, J. Liu, X. Lin, Q. Wu, W. Susilo, SS-DID: A Secure and Scalable Web3 Decentralized Identity Utilizing Multilayer Sharding Blockchain, IEEE Internet Things J 11 (2024) 25694–25705. https://doi.org/10.1109/JIOT.2024.3380068.

[18] D. Vidal-Tomás, The illusion of the metaverse and meta-economy, International Review of Financial Analysis 86 (2023). https://doi.org/10.1016/j.irfa.2023.102560.

[19] R. Belk, M. Humayun, M. Brouard, Money, possessions, and ownership in the Metaverse: NFTs, cryptocurrencies, Web3 and Wild Markets, J Bus Res 153 (2022) 198–205. https://doi.org/10.1016/j.jbusres.2022.08.031.

[20] Y. Lu, The blockchain: State-of-the-art and research challenges, J Ind Inf Integr 15 (2019) 80–90. https://doi.org/10.1016/j.jii.2019.04.002.

[21] A. Shahidinejad, J. Abawajy, An All-Inclusive Taxonomy and Critical Review of Blockchain-Assisted Authentication and Session Key Generation Protocols for IoT, ACM Comput Surv 56 (2024). https://doi.org/10.1145/3645087.

[22] V. Varriale, A. Cammarano, F. Michelino, M. Caputo, The role of digital technologies in production systems for achieving sustainable development goals, Sustain Prod Consum 47 (2024) 87–104. https://doi.org/10.1016/j.spc.2024.03.035.

[23] V.J. Morkunas, J. Paschen, E. Boon, How blockchain technologies impact your business model, Bus Horiz 62 (2019) 295–306. https://doi.org/10.1016/j.bushor.2019.01.009.

[24] REGLAMENTO (UE) 2016/ 679 DEL PARLAMENTO EUROPEO Y DEL CONSEJO, 2016.

[25] O. Akanfe, D. Lawong, H.R. Rao, Blockchain technology and privacy regulation: Reviewing frictions and synthesizing opportunities, Int J Inf Manage 76 (2024). https://doi.org/10.1016/j.ijinfomgt.2024.102753.
[26] Organización de las Naciones Unidas, https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/, (2015).
[27] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008. www.bitcoin.org.
[28] https://ethereum.org/es/, (2025).
[29] A. Yakovenko, Solana: A new architecture for a high performance blockchain v0.8.13, 2017.
[30] A. Kiayias, A. Russell, B. David, R. Oliynykov, Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol, 2019. https://bitcointalk.org/index.php?topic=27787.0.
[31] D. Larimer, https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md, (2018).
[32] Binance, https://github.com/bnb-chain/whitepaper/blob/master/WHITEPAPER.md, (2021).
[33] Mysten Labs, Sui Blockchain Whitepaper, 2023.
[34] https://www.avax.network/, (2025).
[35] https://polygon.technology/, (2025).
[36] Python Software Foundation, Python, (2024).
[37] https://www.linuxfoundation.org/, (2025).
[38] E. Androulaki, A. Barger, V. Bortnikov, S. Muralidharan, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Murthy, C. Ferris, G. Laventman, Y. Manevich, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S.W. Cocco, J. Yellick, Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains, in: Proceedings of the 13th EuroSys Conference, EuroSys 2018, Association for Computing Machinery, Inc, 2018. https://doi.org/10.1145/3190508.3190538.
[39] R.G. Brown, J. Carlyle, I. Grigg, M. Hearn, Corda: An Introduction, 2016.
[40] JP Morgan, https://github.com/Consensys/quorum/blob/master/docs/Quorum%20Whitepaper%20v0.2.pdf , (2018).
[41] Hyperledger Foundation, https://besu.hyperledger.org/, (2023).
[42] B. Chase, E. MacBrough, Analysis of the XRP Ledger Consensus Protocol, (2018). http://arxiv.org/abs/1802.07242.
[43] K. Olson, M. Bowman, J. Mitchell, S. Amudson, D. Middleton, C. Montgomery, Sawtooth: An Introduction, 2018.
[44] G. Wood, POLKADOT: VISION FOR A HETEROGENEOUS MULTI-CHAIN FRAMEWORK, 2016. https://github.com/ethereum/wiki/wiki/Chain-Fibers-Redux.
[45] VeChain Foundation, DEVELOPMENT PLAN AND WHITEPAPER, 2018.
[46] IOTA Foundation, https://www.iota.org/, (2025).
[47] Z. Zheng, S. Xie, H.N. Dai, W. Chen, X. Chen, J. Weng, M. Imran, An overview on smart contracts: Challenges, advances and platforms, Future Generation Computer Systems 105 (2020) 475–491. https://doi.org/10.1016/j.future.2019.12.019.
[48] T. Hewa, M. Ylianttila, M. Liyanage, Survey on blockchain based smart contracts: Applications, opportunities and challenges, Journal of Network and Computer Applications 177 (2021). https://doi.org/10.1016/j.jnca.2020.102857.
[49] W. Zou, D. Lo, P.S. Kochhar, X.B.D. Le, X. Xia, Y. Feng, Z. Chen, B. Xu, Smart Contract Development: Challenges and Opportunities, IEEE Transactions on Software Engineering 47 (2021) 2084–2106. https://doi.org/10.1109/TSE.2019.2942301.
[50] T. Hu, X. Liu, T. Chen, X. Zhang, X. Huang, W. Niu, J. Lu, K. Zhou, Y. Liu, Transaction-based classification and detection approach for Ethereum smart contract, Inf Process Manag 58 (2021). https://doi.org/10.1016/j.ipm.2020.102462.
[51] Winston W. Royce, Managing the Development of Large Software Systems, 1970.

[52] W. Serrano, Verification and Validation for data marketplaces via a blockchain and smart contracts, Blockchain: Research and Applications 3 (2022). https://doi.org/10.1016/j.bcra.2022.100100.
[53] Project Management Institute (PMI), A Guide to the Project Management Body of Knowledge (PMBOK Guide) – Sixth Edition, 2017.
[54] A. Gholam, Blockchain as a Project Management Tool, 2021.
[55] https://www.blockchainresearchinstitute.org/, (2025).
[56] International Project Management Association, Individual Competence Baseline for Project, Programme & Portfolio Management (ICB Version 4.0), 2015.
[57] F. Al Ali, F. Al Shehhi, A. Shamayleh, D. Abuhalimeh, THE APPLICATION OF BLOCKCHAIN TECHNOLOGY IN PROJECT MANAGEMENT, in: Proceedings of International Structural Engineering and Construction, ISEC Press, 2023: p. CPM-08-1-CPM-08-6. https://doi.org/10.14455/ISEC.2023.10(1).CPM-08.
[58] R. Perera, R. Wickramarachchi, C. Rajapakse, Applications of Blockchain Technology in Project Management - A Systematic Literature Review, in: 2023 2nd International Conference for Innovation in Technology, INOCON 2023, Institute of Electrical and Electronics Engineers Inc., 2023. https://doi.org/10.1109/INOCON57975.2023.10101235.
[59] S. Dong, K. Abbas, M. Li, J. Kamruzzaman, Blockchain technology and application: an overview, PeerJ Comput Sci 9 (2023). https://doi.org/10.7717/peerj-cs.1705.
[60] J.P. Womack, D.T. Jones, Lean Thinking—Banish Waste and Create Wealth in your Corporation, Journal of the Operational Research Society 48 (1997) 1148–1148. https://doi.org/10.1038/sj.jors.2600967.
[61] F. Spychiger, M. Lustenberger, J. Martignoni, L. Schädler, P. Lehner, Organizing projects with blockchain through a decentralized autonomous organization (DAO), Project Leadership and Society 4 (2023). https://doi.org/10.1016/j.plas.2023.100102.
[62] J.K. Liker, Toyota Way: 14 Management Principles from the World's Greatest Manufacturer, 1st Edition, McGraw-Hill Education, New York, 2004. https://www.accessengineeringlibrary.com/content/book/9780071392310.
[63] B. Najafi, A. Najafi, A. Farahmandian, The Impact of Artificial Intelligence and Blockchain on Six Sigma: A Systematic Literature Review of the Evidence and Implications, IEEE Trans Eng Manag 71 (2024) 10261–10294. https://doi.org/10.1109/TEM.2023.3324542.
[64] D. Florian, S. Quan Z., M. Hamid, Business Process Management Workshops, Springer International Publishing, Cham, 2018. https://doi.org/10.1007/978-3-030-11641-5.
[65] G. Perboli, S. Musso, M. Rosano, Blockchain in Logistics and Supply Chain: A Lean Approach for Designing Real-World Use Cases, IEEE Access 6 (2018) 62018–62028. https://doi.org/10.1109/ACCESS.2018.2875782.
[66] https://agilemanifesto.org/iso/es/principles.html, (2025).
[67] J. Udvaros, N. Forman, S.M. Avornicului, Agile Storyboard and Software Development Leveraging Smart Contract Technology in Order to Increase Stakeholder Confidence, Electronics (Switzerland) 12 (2023). https://doi.org/10.3390/electronics12020426.
[68] M.S. Farooq, Z. Kalim, J.N. Qureshi, S. Rasheed, A. Abid, A Blockchain-Based Framework for Distributed Agile Software Development, IEEE Access 10 (2022) 17977–17995. https://doi.org/10.1109/ACCESS.2022.3146953.
[69] V. Lenarduzzi, M.I. Lunesu, M. Marchesi, R. Tonelli, Blockchain applications for Agile methodologies, in: ACM International Conference Proceeding Series, Association for Computing Machinery, 2018. https://doi.org/10.1145/3234152.3234155.
[70] https://app.diagrams.net/, (2025).
[71] Microsoft Excel, (2025).
[72] GitHub, https://github.com, (2025).
[73] Daniel Fernández López, https://github.com/danitoo8/MasterThesis, (2025).
[74] Ethereum Foundation, https://remix.ethereum.org, (2025).
[75] https://etherscan.io/, (2025).

[76] Meta, https://react.dev/, (2025).
[77] Ethereum Foundation, https://sepolia.dev/, (2025).
[78] MetaMask, https://metamask.io/, (2025).
[79] Git-scm.com, Git for Windows (Git Bash), (2023).
[80] J. Miguel Ibañez de Aldecoa Quintana, NIVELES DE MADUREZ DE LA TECNOLOGÍA. TECHNOLOGY READINESS LEVELS. TRLS, n.d.
[81] https://www.sap.com/spain/index.html, (2025).