

**MÁSTER UNIVERSITARIO EN INVESTIGACIÓN EN INGENIERÍA DE SOFTWARE Y
SISTEMAS INFORMÁTICOS**



(31105151)

[Trabajo Fin de Máster en Ingeniería de Software y Sistemas Informáticos](#)

**Aplicación de las áreas prácticas del estándar CMMI DEV 2.0 Nivel 2 en una PYME
de Consultoría Tecnológica**

Alumna: Elisa Parejo Vadillo

Director: María Magdalena Arcilla Cobián

Curso académico 25/26

Convocatoria de defensa: Septiembre

**MÁSTER UNIVERSITARIO EN INVESTIGACIÓN EN INGENIERÍA DE SOFTWARE Y
SISTEMAS INFORMÁTICOS**



(31105151)

[Trabajo Fin de Máster en Ingeniería de Software y Sistemas Informáticos](#)

**Aplicación de las áreas prácticas del estándar CMMI DEV 2.0 Nivel 2 en una PYME
de Consultoría Tecnológica**

Alumna: Elisa Parejo Vadillo

Director: María Magdalena Arcilla Cobián

Curso académico 25/26

Convocatoria de defensa: Septiembre

Resumen

La creciente dependencia del software en los procesos industriales, especialmente en el contexto de la Industria 4.0, ha impulsado a las consultoras tecnológicas a desarrollar soluciones ad-hoc, enfrentándose a desafíos de predictibilidad en plazos, costes y calidad. Sin embargo, la adopción de modelos de madurez como CMMI-DEV V2.0 en su Nivel de Madurez 2 (ML2) resulta inviable para micropymes debido a la elevada carga burocrática, los costes de certificación y la fricción con metodologías ágiles como Scrum.

Esta investigación aborda la pregunta: ¿cómo aplicar y automatizar el modelo CMMI-DEV V2.0 ML2 en una PYME de consultoría tecnológica que desarrolla software ad-hoc, mejorando la predictibilidad sin incurrir en una carga burocrática insostenible? Para dar respuesta, se diseña un modelo híbrido conceptual que traduce las exigencias del Área de Práctica RDM (*Requirements Development and Management*) de CMMI DEV V2.0 en su nivel de madurez 2 al ecosistema nativo de Scrum. Para ello establece correspondencias sistemáticas entre las prácticas del estándar y los artefactos ágiles de Scrum.

Como aportación práctica principal, se ha desarrollado una Prueba de Concepto (PoC) que consiste en un motor de auditoría automatizado en Python con arquitectura FastAPI, persistencia SQLite y análisis mediante expresiones regulares sobre repositorios Git con archivos Markdown. El sistema escanea, valida y genera reportes que presentan el grado de conformidad con la aplicación de las prácticas RDM. El sistema se diseña bajo la premisa clave de introducir una arquitectura modular que permita para abordar de forma escalonada un proyecto de implantación del estándar en entornos de micropymes. Esta extensibilidad se demuestra incorporar nuevas áreas de práctica mediante la modificación de un manifiesto de configuración externo.

La validación experimental se ha realizado sobre un repositorio piloto que confirma la viabilidad de la solución: La herramienta reduce de forma drástica la carga manual de auditoría, garantiza la trazabilidad bidireccional (del requisito al resultado y del resultado al requisito), y proporciona evidencias objetivas que sirven como base para para futuras evaluaciones formales. El trabajo pone de manifiesto que es posible integrar la rigurosidad normativa de CMMI con la flexibilidad ágil mediante un motor de automatización que facilita una vía accesible y escalable para que micropymes

tecnológicas mejoren su madurez organizativa sin sacrificar su dinámica operativa ni incurrir en inversiones prohibitivas.

Palabras clave

CMMI DEV V2.0, Nivel de Madurez 2, Scrum, Auditoría Automatizada, Gobierno TI, Trazabilidad de Requisitos, Expresiones Regulares, FastAPI, Python.

Índice

Resumen	5
Palabras clave	6
CAPÍTULO 1. Introducción.....	13
1. Introducción	14
1.1. Objetivos del TFM.....	15
1.1.1. Objetivo General	15
1.1.2. Objetivos Específicos	15
1.1.3. Estructura del TFM	16
CAPÍTULO 2. Justificación del problema y Pregunta de Investigación.....	18
2. Justificación del problema y Pregunta de Investigación.....	19
2.1. Alcance y Limitaciones del Proyecto.	22
2.1.1. Alcance del Proyecto.....	22
2.1.2. Limitaciones del Proyecto.....	23
CAPÍTULO 3. Marco Teórico y Estado de la cuestión	24
3. Marco teórico y Estado de la cuestión.....	25
3.1. Evolución y estructura del modelo CMMI: bases para la mejora de procesos	26
3.1.1. De SW-CMM a CMMI V1.3. El enfoque de constelaciones.....	26
3.1.2. CMMI V2.0 e ISACA: Rendimiento, valor de negocio y agilidad.....	27
3.1.3. CMMI v3.0 y la extensión AIM (Inteligencia Artificial).....	27
3.2. Diseño metodológico de CMMI: Enfoques por Capacidad y Madurez.	28
3.3. CMMI V2.0: Rendimiento, valor de negocio y agilidad.....	31
3.3.1. Arquitectura de Componentes	32
3.3.2. Niveles de Madurez vs Niveles de Capacidad en CMMI V2.0	32
3.3.3. Transformación Estructural De Constelaciones a Vistas.	34
3.3.4. Integración de prácticas de seguridad en su estructura principal.....	36
3.4. Implicaciones para la Mejora de Procesos	36
3.4.1. Reorientación estratégica del modelo de la institucionalización al rendimiento:	36
3.5. Cambio de modelo a Producto: Implementación y Evaluación:.....	38
3.6. CMMI Development V2.0.....	40
3.6.1. CMMI DEV V2.0 NIVEL DE MADUREZ 2	45
CAPITULO 4 . Propuesta del modelo conceptual	50
4. Propuesta del Modelo Conceptual	51
4.1. Introducción	51
4.2. Análisis de Contexto.....	51
4.3. Estructura y dinámica de trabajo	52
4.4. Motivación para la adopción de CMMI	53
4.5. Análisis de Brecha (Gap Analysis).....	54
4.5.1. Metodología de evaluación.....	54
4.6. Mapeo metodológico y Matriz de trazabilidad cruzada:	75

4.6.1.	Análisis de la Matriz de Relación PA CMMI DEV V2.0 ML2-Artefacto Scrum	75
4.7.	Roles y responsabilidades en el modelo híbrido	78
4.7.1.	Definición de roles	78
4.7.2.	Matriz de casos de uso (modelo conceptual)	78
4.7.3.	Matriz RACI.....	80
CAPITULO 5 . Desarrollo técnico y validación.....		82
5.	Desarrollo Técnico y Validación:	83
5.1.	Descripción de alto nivel	83
5.2.	Stack Tecnológico	84
5.3.	Requisitos del sistema	85
5.3.1.	Requisitos No funcionales:	85
5.3.2.	Requisitos funcionales	88
5.4.	Descripción funcional:	93
5.4.1.	Arquitectura funcional:.....	93
5.5.	Desarrollo de la Prueba de Concepto (PoC)	106
5.5.1.	Arquitectura de despliegue.....	106
5.5.2.	El manifiesto de configuración	107
5.5.3.	Refactorización Arquitectónica para Modularidad.....	108
5.5.4.	Demostración Práctica de la Modularidad.....	110
5.5.5.	Resultados de la Validación	114
CAPITULO 6 . Conclusiones y líneas futuras		116
6.	Conclusiones y Líneas futuras.....	117
6.1.	Evaluación del Cumplimiento de los Objetivos.....	118
6.1.1.	Cumplimiento Objetivo General	118
6.1.2.	Cumplimiento de Objetivos Específicos	118
6.1.3.	Contribuciones Principales del Trabajo	121
6.2.	Lecciones aprendidas	122
6.2.1.	Sobre el proceso de adopción de CMMI en micropymes.	123
6.2.2.	Sobre el desarrollo de la PoC	123
6.3.	Limitaciones del Estudio	124
6.4.	Líneas de Trabajo Futuro	125
6.4.1.	Extensión a Otras Áreas de Práctica de CMMI	125
6.4.2.	Mejoras Técnicas y Arquitectónicas.....	125
CAPITULO 7 . Bibliografía y Referencias		127
7.	Bibliografía y referencias	128
LISTADOS.....		133
Listado de Siglas		134
ANEXOS.....		139
Anexo I. Diseño de la pre-evaluación interna:.....		140
Anexo II Matriz de relación prácticas CMMI V2.0.....		149

Anexo III: Matriz Prácticas CMMI DEV V2.0.	161
Anexo IV Matriz de relación áreas de prácticas -artefectos Scrum, para la definición de Requisitos funcionales ..	170

Índice de Ilustraciones:

Ilustración 1. Línea temporal de evolución modelo CMMI. Elaboración propia	28
Ilustración 2. Arquitectura integral de componentes en CMMI V1.3. Elaboración propia.	35
Ilustración 3. Arquitectura integral modular en CMMI V2.0. elaboración Propia	36
Ilustración 4. Diagrama de flujo de adopción CMMI V2.0. Elaboración Propia	40
Ilustración 5. Estructura CMMI DEV V2.0. Elaboración propia.....	42
Ilustración 6. PAs de Evaluación CMMI DEV 2.0 ML2. Elaboración propia	45
Ilustración 7. Mapa de calor niveles madurez CMMI DEV V2.0. Elaboración propia	49
Ilustración 8. Arquitectura lógica modular basada en componentes	84
Ilustración 9. Modelo de Kruchten. Elaboración propia	93
Ilustración 10. Vista Física. Diagrama de despliegue. elaboración propia	96
Ilustración 11. Vista de desarrollo. Diagrama de componentes. Elaboración propia.....	97
Ilustración 12. Vista Lógica. Diagrama de clases. elaboración propia	98
Ilustración 13. Vista de Procesos. Diagrama de Actividad. Elaboración propia	99
Ilustración 14. Vista de Casos de Uso. Elaboración propia.....	100
Ilustración 15. Estructura de archivos despliegue cmimi-auditor poc.....	106
Ilustración 16. Dashboard con configuración inicial Proyecto SCM (RDM)	110
Ilustración 17. Dashboard con configuración inicial Proyecto MIVELOCIDAD3 (rdm)	111
Ilustración 18. dashboard para configuración modular	112
Ilustración 19. Dashboard vista de detalle módulo RDM en configuración multi-módulo	113
Ilustración 20. Informe impreso desde dashboard resultado evaluación	114

Índice de Tablas:

Tabla 1. Niveles de Capacidad Vs Niveles de madurez CMMI. Adaptado del manual CMMI Institute.....	29
Tabla 2. Matriz niveles capacidad vs niveles de madurez	31
Tabla 3. Jerarquía Estructural CMMI V2.0. Adaptada de (CMMI Institute, 2019)	32
Tabla 4. Niveles de capacidad adaptados CMMI V2.0	33
Tabla 5. Niveles de Madurez CMMI adaptado V2.0	34
Tabla 6. Criterio de Comparación Enfoque de ROI en CMMI V1.3 vs CMMI V2.0	37
Tabla 7. Atributos esenciales estructura V1.3 vs V2.0	38
Tabla 8. Resumen cuantitativo de la estructura CMMI DEV 2.0	41
Tabla 9. Resumen Cualitativos Prácticas desglose por nivel de madurez cmimi dev extendido 2.0. Adaptada de (CMMI Institute, 2018)	44
Tabla 10. Resumen cuantitativo de prácticas CMMI DEV V2.0 ML2	46
Tabla 11. PAs por nivel CMMI DEV 2.0 ML2. Bloque 1	46
Tabla 12. PAs por nivel CMMI DEV 2.0 ML2. Bloque 2	47

Tabla 13. PAs por nivel CMMI DEV 2.0 ML2. Bloque 3	47
Tabla 14. PAs por nivel CMMI DEV 2.0 ML2. Bloque 4	47
Tabla 15. PAs por nivel CMMI DEV 2.0 ML2. Resumen	48
Tabla 16. Listado de prácticas por PA's CMMI DEV V2.0 ML2.....	49
Tabla 17. Criterios de calificación adaptados de CMMI V2.0.	56
Tabla 18. Resumen Resultado Pre evaluación CMMI DEV 2.0 ML2	57
Tabla 19. Resultados auditoria CMMI DEV V2.0 ML2	58
Tabla 20. Clasificación de brechas por severidad	71
Tabla 21. Contraste entre declaraciones y evidencias	72
Tabla 22. Plan de acción priorización.....	73
Tabla 23. Relación prácticas CMMI DEV V2.0 ML2-Artefactos SCRUM. Adaptada (Henriquez et al., 2021, 2022).....	77
Tabla 24. Definición de Roles modelo híbrido.....	78
Tabla 25. Matriz casos de uso modelo conceptual	80
Tabla 26. Matriz RACI	81
Tabla 27. Requisitos No Funcionales según las Perspectivas de McCall	87
Tabla 28. Trazabilidad entre RF, prácticas CMMI y archivos de código.....	88
Tabla 29. Matriz Casos Uso. Enfoque CMMI DEV 2.0 ML2 RDM	105
Tabla 30. Archivos refactorizados para modularidad	109
Tabla 31. Cumplimiento de objetivos específicos.....	121

CAPÍTULO 1. Introducción

1. Introducción

El sector industrial ha transformado sus operaciones con la incorporación de tecnologías habilitadoras del paradigma de la industria 4.0. y la integración de sistemas (Arjun & Smitha, 2025; Villar et al., 2024). En este nuevo paradigma, la manufactura y el desarrollo de software van intrínsecamente unidos. La eficiencia de los sistemas ciberfísicos está supeditada a la madurez del software que la gestiona (Villar et al., 2024), lo que lo convierte en motor fundamental de la innovación, la competitividad y la eficiencia operativa de todos los sectores industriales (Giordano et al., 2024) (Palomeque-Jaramillo et al., 2024).

Esta dependencia exige que la industria del software adopte procesos predecibles y escalables. Sin embargo, en los equipos de desarrollo de software, es un problema recurrente la desviación en los costes y en los plazos de entrega y calidad del producto (Alves et al., 2021).

Los procesos de fabricación de bienes físicos, en cambio, se han depurado mediante estándares que garantizan la repetibilidad del producto. En esos sectores, la infraestructura tecnológica tiene un impacto crítico y superior al talento: la maquinaria y los sistemas de automatización establecen los límites de la capacidad productiva, la velocidad y la precisión. En el desarrollo de software, en cambio, la tecnología de soporte está más centrada en la naturaleza del desarrollo, y el talento del equipo resulta determinante. Para reducir esta dependencia, conviene adoptar modelos de mejora que proporcionen un marco de trabajo común (Libo, 2025), con un enfoque basado en evidencias (Kitchenham et al., 2004), que integre métricas de predictibilidad y estandarice el ciclo de vida con gobernanza robusta alineada con los niveles de madurez de CMMI (*Capability Maturity Model Integration*). Este enfoque reduce la incertidumbre y hace que los resultados sean consistentes con independencia del entorno.

Sin embargo, las metodologías de gestión de proyectos encuentran dificultades de arranque en cualquier organización, de forma general el cambio de modelo de gestión exige una curva de aprendizaje muy pronunciada y un proceso de

adaptación metodológico a largo plazo (Pussella & Bandara, 2018). En las empresas pequeñas, es necesario reducir estos tiempos y adoptar estructuras organizativas ágiles.

A pesar de sus ventajas, implantar CMMI entra en conflicto con la flexibilidad de las metodologías ágiles como Scrum., que predominan en la industria actual. Esto se debe a la alta carga burocrática y el coste en recursos que precisa la verificación manual de evidencias. Este es el contexto en el que se enmarca el presente Trabajo Fin de Máster. Para resolver la brecha entre ambos enfoques, esta investigación propone un modelo híbrido conceptual, que traduce las exigencias del Área de Práctica (*Practice Area*) de RDM de CMMI V2.0 ML2 al entorno nativo de Scrum. Como principal contribución práctica, se desarrolla un motor de software ad-hoc en Python con arquitectura FastAPI que se encarga de escanear y auditar, de forma automatizada y mediante expresiones regulares (regex), la conformidad y trazabilidad de los artefactos Scrum, lo que facilita un gobierno de TI continuo y objetivo

1.1. Objetivos del TFM

1.1.1. Objetivo General

Diseñar, implementar y validar un sistema de auditoría automatizada para un enfoque híbrido Agile Scrum-CMMI DEV V2.0 ML2., para ello se desarrollará una herramienta software ad-hoc en Python que permita el control de la calidad del proceso de desarrollo de software, verifique la trazabilidad y audite la conformidad de los entregables.

1.1.2. Objetivos Específicos

OE 1. Estudio del estado del arte: Investigar la evolución histórica del modelo CMMI y la estructura detallada de su versión V2.0, y la relación con las metodologías ágil Scrum, analizando la literatura académica existente. Este análisis permitirá identificar cómo coexisten ambos enfoques y qué limitaciones tienen las soluciones actuales para auditarlos de forma automatizada

OE 2. Diseño del modelo híbrido conceptual: Mapear las áreas de práctica de CMMI V2.0 ML2 (en concreto RDM) con los eventos y artefactos documentales de Scrum, para definir la estructura que deben tener los artefactos documentales del repositorio y los patrones de sintaxis (expresiones regulares) que permiten demostrar el cumplimiento normativo.

OE 3. Desarrollo de una herramienta en Python: Construir un motor de software en Python con arquitectura API REST (FastAPI). La herramienta tiene que escanear repositorios de Git locales donde se almacenan los proyectos, analizar archivos Markdown mediante expresiones regulares para verificar la trazabilidad bidireccional (del requisito al resultado y del resultado al requisito), validar reglas CMMI DEV V2.0 ML2 RDM, y generar un reporte de auditoría con persistencia en la base de datos (SQLite).

OE 4. Validación experimental: Desplegar la solución en un proyecto piloto simulado, a lo largo de varios Sprints, para extraer métricas y generar automáticamente un informe de evidencias de auditoría CMMI.

1.1.3. Estructura del TFM

El TFM se organiza en seis capítulos:

Capítulo 1. Introducción y Objetivos. Introduce el contexto de la Industria 4.0 en el que se sitúa la investigación. En esta sección se definen las metas del trabajo y se detallan el objetivo general y los objetivos específicos establecidos.

Capítulo 2. Justificación del Problema: Formula formalmente el problema de investigación, planteando los desafíos que presenta la adopción de CMMI-DEV V2.0 ML2 en micropymes de consultoría industrial que desarrollan software ad-hoc en el contexto de la I4.0. El bloque cierra con la formulación pregunta central de investigación.

Capítulo 3. Marco Teórico y Estado del Arte: Revisa la literatura sobre modelos de mejora de procesos, la evolución de CMMI y las limitaciones actuales para automatizar auditorías. Por otro lado, establece un marco de relación entre los artefactos de Scrum las PA del estándar CMMI DEV V2.0.

Capítulo 4. Propuesta del Modelo Conceptual: Desarrolla el modelo conceptual del trabajo. En él se describe el entorno empresarial de aplicación de micropymes y se expone el análisis de brecha (*gap analysis*) que se realiza a través de un modelo de preevaluación diseñado de acuerdo con el estándar CMMI V2.0. Por último, se presentan los resultados obtenidos.

Capítulo 5. Desarrollo Técnico y Validación: Expone el diseño de la herramienta de software programada en Python (FastAPI, SQLite y expresiones regulares) para escanear repositorios Git. Así mismo, se detallan las métricas obtenidas de la validación experimental de los casos pilotos auditados por la herramienta.

Capítulo 6. Conclusiones: sintetiza los hallazgos, evalúa el cumplimiento de los objetivos y propone líneas de trabajo futuro.

Capítulo 7. Bibliografía y referencias: "Recoge las fuentes académicas, los documentos oficiales de CMMI, los artículos de investigación y las publicaciones especializadas que sustentan el trabajo.

CAPÍTULO 2. Justificación del problema y Pregunta de Investigación

2. Justificación del problema y Pregunta de Investigación

La adopción de modelos de madurez como CMMI-DEV V2.0 en su Nivel de Madurez 2 (ML2) ha sido históricamente un privilegio exclusivo de grandes corporaciones de software, debido a los altos costes de consultoría y certificación, la complejidad burocrática y documental, el elevado precio de las herramientas ALM (Application Lifecycle Management) empresariales y la necesidad de disponer de personal dedicado exclusivamente a la gestión de procesos.

En el tejido empresarial actual, las micropymes de desarrollo tecnológico. se enfrentan a una doble problemática: por un lado, la necesidad de operar bajo estándares de calidad exigidos por clientes o licitaciones; por otro, la carencia de presupuestos para herramientas y personal dedicado a la gestión de procesos. Existe, por tanto, una brecha crítica en el mercado: no disponen de una solución metodológica ni técnica que fusione la agilidad diaria (Scrum) con el control riguroso de CMMI V2.0 de forma gratuita y automatizada.

A pesar de los beneficios de la implementación del CMMI, su éxito está fuertemente condicionado por el tamaño de la organización. Las empresas grandes tienen capacidad de trabajar bajo modelos probados, mientras que las empresas pequeñas tienen amplias dificultades, en parte por la ausencia de estructura y en parte por limitaciones presupuestarias para absorber los costes de implantación y certificación (Bradac Hojnik et al., 2023; Cukier, 2008). Esta realidad se agrava el contexto de la consultoría industrial, donde la irrupción masiva del software en el control de los procesos de manufactura, ha obligado a situar la digitalización como eje central de la mejora en las empresas de sus clientes. Los procesos operativos están diseñados de forma específica para un entorno concreto, por lo que las soluciones comerciales, no cubren las necesidades, lo que ha abocado a las consultoras al desarrollo de software ad-hoc destinado a digitalizar procesos operativos. Que han tenido que modificar su propuesta de valor hacia la consultoría tecnológica, impulsando la mejora de los procesos de manufactura a través de la introducción de herramientas digitales, donde el conocimiento del entorno y los puntos críticos de los procesos de manufactura son el requisito indispensable para la adecuación de soluciones basadas en software (Happonen et al., 2020).

Sin embargo, el *core* de estas organizaciones no procede de la ingeniería de desarrollo, por ello carecen de procesos maduros, lo que se traduce en sobrecostos, retrasos y calidad variable en los entregables softwares. Para abordar esta carencia, los modelos de madurez de procesos como CMMI ofrecen una guía contrastada para abordar el cambio. En particular, el nivel 2 (Gestionado) de CMMI DEV V2.0 se articula en torno a un conjunto de áreas de proceso que resultan fundamentales para mejorar la predictibilidad en la ejecución de proyectos, pues permiten a cualquier organización sin importar su tamaño controlar los plazos, los costes y la calidad de sus entregables. Aunque CMMI ha demostrado su eficacia en grandes organizaciones y en empresas de software puras, su adopción en una pyme requiere ajustar las prácticas al tamaño reducido del equipo con el objetivo de no obstaculizar el avance en los proyectos de cliente y superar las barreras que pueden plantearse en la transición desde un enfoque tradicional de desarrollo hacia un enfoque ágil en entornos de pequeño tamaño (Pussella & Bandara, 2018).

Históricamente, la demostración del cumplimiento de CMMI se ha asociado a procesos burocráticos, estáticos y con alta carga de documentación manual. Por el contrario, Scrum promueve la priorización del "software funcionando sobre la documentación exhaustiva" (The Agile Manifesto Authors, 2001). Al intentar implantar este modelo híbrido, las micropymes se enfrentan a tres desafíos críticos no resueltos por las soluciones comerciales actuales:

1. Inviabilidad de la Auditoría Continua: La verificación del cumplimiento de las prácticas de CMMI se realiza de forma retrospectiva y manual por parte de auditores o miembros de los equipos de calidad. En un entorno Scrum de despliegue rápido, no resulta viable, es costoso por la necesidad de recursos asociados y facilita la comisión de errores. Esta situación deriva en un cuello de botella que ralentiza los Sprints, degradando la naturaleza ágil del entorno.
2. Pérdida de la Trazabilidad Bidireccional: Uno de los puntos más críticos de la Practice Area de Desarrollo y Gestión de Requisitos (RDM) de CMMI es garantizar que cada necesidad de los stakeholders esté perfectamente conectada con su código y criterios de aceptación. Sin embargo, a causa de la velocidad de los cambios en los repositorios ágiles, esta trazabilidad suele

romperse o desactualizarse. Como consecuencia, se produce una pérdida de control sobre el alcance real del proyecto.

3. Subjetividad en la Validación de Entregables: Toda validación de criterios en ingeniería del software, necesita establecer estándares de comparación que garanticen su objetividad. Evaluar la conformidad los artefactos del proyecto, como el *Product Backlog* o la *Definition of Done* (DoD), respecto a un estándar de empresa resulta un proceso subjetivo, especialmente en entornos de subcontratación o equipos distribuidos, que responde a la realidad de las empresas de pequeño tamaño. Actualmente no existen herramientas ligeras y automáticas que actúen como un filtro de calidad inmediato antes de validar una entrega.

Ante este escenario, la presente investigación se justifica por la ausencia de soluciones tecnológicas ad-hoc y de bajo coste capaces de realizar un análisis estático y continuo de los repositorios de software para medir la conformidad con los estándares. El desarrollo de este motor automatizado en Python no solo resuelve una brecha técnica mediante el uso eficiente de expresiones regulares, sino que aporta un beneficio metodológico directo: demuestra que es posible introducir la gobernanza de forma escalonada un proceso de evolución continua hacia un nivel de madurez organizativa riguroso (CMMI ML2) sin sacrificar la agilidad, autonomía y dinamismo propios del marco Scrum.

Por todo ello, este Trabajo Fin de Máster se plantea la siguiente pregunta de investigación: ¿Cómo aplicar y automatizar el modelo CMMI DEV V2.0 de Nivel de Madurez 2 en una micropyme de consultoría industrial que ha transformado su propuesta de valor desde el desarrollo de proyectos basados en metodologías clásicas que para el desarrollo de la excelencia operacional hacia la consultoría tecnológica y el desarrollo de software ad-hoc, de forma que mejore la predictibilidad en sus plazos, costes y calidad, sin incurrir en una carga burocrática inviable para su tamaño y naturaleza?

2.1. Alcance y Limitaciones del Proyecto.

2.1.1. Alcance del Proyecto.

El alcance de este Trabajo Fin de Máster comprende el diseño metodológico y el desarrollo técnico de una solución de auditoría automatizada adaptada a micropymes tecnológicas. Los límites de la implementación práctica se definen bajo los siguientes componentes:

- **Área de Proceso Seleccionada:** La solución se acota estrictamente a la *Practice* Área de Desarrollo y Gestión de Requisitos (RDM - Requirements Development and Management) del modelo CMMI-DEV V2.0, enfocándose en las prácticas correspondientes a los Niveles 1 y 2 de madurez. La decisión de implementar esta área de practica, y no el resto de las diez áreas que son necesarias para alcanzar un nivel de madurez 2 en la organización, se fundamenta en los resultados de la preevaluación diseñada y ejecutada al efecto de encontrar una guía de implementación en la empresa a futuro.
- **Marco Ágil de Trabajo:** El motor está diseñado de manera exclusiva para proyectos gestionados bajo el marco Scrum. El análisis se centra en la estructura nativa de sus artefactos principales: el *Product Backlog*, articulado mediante historias de usuario (*User Stories*), los compromisos del Sprint (*Sprint Commitments*) y la *DoD*.
- **Stack tecnológico de la herramienta:** El desarrollo comprende un motor de software *ad-hoc* programado en Python 3.14. Su arquitectura se compone de un núcleo de análisis por expresiones regulares, un sistema de persistencia de datos local en SQLite y una interfaz API REST construida con FastAPI para su consumo e integración continua.
- **Validación Experimental:** El alcance de la prueba piloto se limita a la auditoría automatizada y recolección de métricas sobre un entorno piloto basado en un caso real de estudio, administrado mediante archivos Markdown (.md), del proyecto *SCM-SISTEMA-MEJORA-CONTINUA* y del proyecto MIVELOCIDAD3.

2.1.2. Limitaciones del Proyecto

Para garantizar la viabilidad del TFM dentro de los plazos académicos, se establecen las siguientes exclusiones y restricciones técnicas:

- Exclusión de otras áreas de proceso CMMI: El software sólo evalúa el área descrita en el apartado anterior. Aunque la arquitectura es extensible, quedan fuera de este trabajo el resto de PA del modelo.
- Análisis sintáctico, no semántico: El motor realiza una auditoría basada en la detección de evidencias, verificación de estructuras de archivos y patrones de texto específicos mediante expresiones regulares (Regex). Queda fuera de su alcance el procesamiento de lenguaje natural (PLN) así como el análisis semántico o cualitativo del contenido.
- Dependencia del Formato Markdown: La herramienta está optimizada y parametrizada para escanear repositorios ligeros que utilicen Markdown como estándar documental. No realiza lectura de otros soportes documentales tales como Word o Excel, ni realiza sincronización automatizada mediante conectores nativos a herramientas ALM propietarias.
- Entorno de Persistencia Local: El almacenamiento se limita a una base de datos relacional ligera y local (SQLite). No se realiza el despliegue en infraestructuras distribuidas en nube o servidores corporativos dedicados.

CAPÍTULO 3. Marco Teórico y Estado de la cuestión

3. Marco teórico y Estado de la cuestión.

Nota metodológica sobre el uso de la terminología oficial: Los conceptos, definiciones y nomenclaturas descritos en este capítulo hacen referencia directa al marco normativo cerrado establecido en el modelo conceptual CMMI® Con el objetivo de preservar la fidelidad técnica, la integridad del modelo y la validez del posterior mapeo metodológico, se han mantenido los términos y definiciones estandarizados (Carnegie Mellon University, 2010; CMMI Institute, 2018). No obstante, las interpretaciones, análisis críticos, propuestas de hibridación y la síntesis de los contenidos presentados en este documento constituyen un desarrollo analítico original de autoría propia.

Dentro de la ingeniería del software, el aseguramiento de calidad se divide en dos aspectos diferenciados, el control sobre la calidad del producto final y la intervención sobre el proceso de desarrollo. (Chrissis et al., 2011). Las metodologías orientadas al producto, se apoyan en mecanismos de inspección directas. Entre ellos destacan la ejecución de baterías de pruebas técnicas, auditorías de código fuente o la aplicación de métricas bajo el estándar ISO/IEC 25000 (FunSoft2020, 2020). Estas medidas son indispensables para evitar el despliegue de versiones con fallos no deseados. Sin embargo sólo pueden realizar después del desarrollo del producto o de alguna de sus fases, y responden a una estrategia reactiva. (Alves et al., 2021). Sin embargo, las metodologías de mejora de procesos son proactivas. Se fundamentan en el control de los procesos y su predictibilidad como base para la excelencia en los resultados (Carnegie Mellon University, 2010). Para el *Software Process Improvement* (SPI), la viabilidad de los proyectos en entornos sin procedimientos estandarizados y consolidados, dependen del talento individual en lugar de la capacidad de la organización (Libo, 2025).

Implementar un enfoque de mejora basado en procesos propicia una gobernanza proactiva. Prácticas como el Desarrollo y Gestión de Requisitos (RDM) aseguran de forma temprana la trazabilidad bidireccional y el entendimiento mutuo con los *stakeholders* (Henriquez et al., 2022) Al estandarizar el "cómo", la organización reduce la variabilidad organizativa e institucionaliza las evidencias objetivas (Kitchenham et al., 2004). Por tanto, el modelo CMMI que se analiza en este capítulo, no sustituye a los controles de técnicos de calidad de software, sino que los dota de

un entorno robusto donde producto final sea el resultado predecible de un proceso maduro.

3.1. Evolución y estructura del modelo CMMI: bases para la mejora de procesos

La metodología CMMI ha evolucionado, de manera transversal consolidando sus áreas de prácticas tanto en el desarrollo de software como en otras disciplinas clave organizacionales como la gestión de servicios o personas. En la actualidad que se integra en las nuevas metodologías ágiles en la gestión de proyectos. Y se presenta como un modelo válido en los procesos con base en la Inteligencia artificial. (CMMI Institute, 2026a) (Reifer, 2000)

3.1.1. De SW-CMM a CMMI V1.3. El enfoque de constelaciones

Dentro de su proceso evolutivo, el modelo CMMI ha cubierto hitos que han supuesto un avance importante en la adaptabilidad del modelo a las necesidades de las empresas demandantes para adaptarse a un enfoque múltiple de manera simultánea.

La aparición de CMMI versión 1.1 en 2002 supone un punto de inflexión crítico en el modelo inicial CMM desarrollado por la universidad de Carnegie Mellon pasando de un enfoque fragmentado por disciplinas, Desarrollo de software SW, ingeniería de sistemas (SE), Adquisiciones (SA), Desarrollo integrado de productos (IPD), a un marco de referencia unificado que elimina las redundancias del modelo inicial.

En la vida del modelo CMMI la versión 1.1, supone la estandarización de prácticas con independencia del dominio de aplicación y una armonización de la terminología de mejora de procesos. A nivel estructural fija las bases de las versiones actuales, que se fundamenta en dos dimensiones, una continua que permite medir la capacidad de los procesos individuales y otra por etapas que evalúa el nivel de madurez de la organización en su conjunto, estructurados en 5 niveles de madurez, desde el nivel 1 (proyectos no predecibles) hasta el nivel 5 (procesos estandarizados, optimizados y en mejora continua), mediante la asimilación de un conjunto homogéneo de prácticas. La importancia de esta versión radica en la perdurabilidad de sus principios estructurales en todas las versiones posteriores.

La versión 1.2 (2006) corrige la arquitectura monolítica mediante una estructura modular, más adecuada para evaluar servicios o compras en vez de desarrollo de ingeniería. En el año 2010, surge el concepto de "Especialización Operativa" y nace formalmente el concepto de "Constelaciones" y el nombre cambia oficialmente a CMMI para Desarrollo (CMMI-DEV V1.3), preparando el terreno para Adquisiciones (2007) y Servicios (2009). Introduce el concepto de "Áreas de Práctica" independientes y guías de contexto como Agile.

3.1.2. CMMI V2.0 e ISACA: Rendimiento, valor de negocio y agilidad.

La versión 2.0 (2018), impulsada por ISACA, dirige en enfoque del modelo hacia el rendimiento del negocio y facilita la incorporación de metodologías ágiles, que encuentran en la estructura unificada del modelo un soporte idóneo para su integración nativa, facilitando su uso en entornos empresariales de pymes o micropymes.

Si bien en la versión 2.0, los componentes están separados a efectos de la evaluación los niveles de madurez siguen vinculados a combinaciones fijas, por tanto, sigue siendo semi acoplado en las reglas de evaluación y actualización.

3.1.3. CMMI v3.0 y la extensión AIM (Inteligencia Artificial)

Actualmente, la incursión de los sistemas de inteligencia artificial, ha generado un nuevo paradigma de desarrollo que plantea riesgos en términos de gobernanza, ética y seguridad. CMMI AIM (2026), nace como un área transversal que impacta en torno al 50% de las áreas de prácticas preexistentes (31 Áreas Práctica), mediante una nueva vista AIM. Esta actualización no altera el Core de la versión anterior, sino que la gestión de los nuevos escenarios y riesgos se apoya sobre los principios de gobierno establecidos en versiones anteriores.



ILUSTRACIÓN 1. LÍNEA TEMPORAL DE EVOLUCIÓN MODELO CMMI. ELABORACION PROPIA

3.2. Diseño metodológico de CMMI: Enfoques por Capacidad y Madurez.

El Modelo de Integración de Modelos de Madurez de Capacidades (CMMI) es un marco metodológico de buenas prácticas y constituye uno de los más consolidados en el ámbito de la ingeniería de software y mejora de los procesos organizacionales. Su principio fundamental sostiene que la calidad de un producto de software es el resultado del seguimiento de un proceso bien estructurado, predecible, desarrollado

con calidad atendiendo a estándares fijados y sometido a ciclos continuos de optimización y control bajo baremos previamente definidos.

A diferencias de otros marcos metodológicos, el modelo CMMI, está diseñado para su implantación práctica y para su evaluación formal (Chrissis et al., 2011).

Aunque en su nacimiento como SW-CMM sólo define el concepto de nivel de madurez, todos los modelos a partir de la primera versión de CMMI, articulan su estructura conforme a dos enfoques independientes pero complementarios entre sí, un enfoque continuo (Capacidad) y un enfoque por etapas (Madurez). Estos dos enfoques, permiten establecer los objetivos de evaluación a nivel micro (procesos individuales) y a nivel macro(la organización en su conjunto) (CMMI Institute, s.f.-a).

La Tabla 1 sintetiza los aspectos más relevantes para medir la capacidad de mejora en una organización

Aspecto	Nivel de Capacidad (Capability Level)	Nivel de Madurez (Maturity Level)
Objeto de evaluación	El desempeño en un Área de Proceso específica	El estado general de mejora de procesos de toda la organización, basado en un conjunto de áreas de proceso.
Enfoque de mejora	Continuo	Por Etapas.
	Mejora selectiva DE PA específicas según objetivos de negocio.	Proporciona una hoja de ruta de mejora predefinida y secuenciada.
Granularidad	Micro	Macro
Grado de adaptabilidad	Optimizar procesos críticos para la organización	Alcanzar un estatus competitivo u homologación formal.

TABLA 1. NIVELES DE CAPACIDAD Vs NIVELES DE MADUREZ CMMI. ADAPTADO DEL MANUAL CMMI INSTITUTE

El modelo CMMI ha mantenido a lo largo de su evolución los conceptos de Nivel de Madurez (ML) y Nivel de Capacidad (CL), aunque su aplicación y nomenclatura han ido evolucionando con el modelo.

Respecto de los niveles de madurez, originalmente, SW-CMM, define cinco niveles de madurez: “Inicial” (1), “Repetible” (2), “Definido” (3), “Gestionado” (4) y “Optimizado” (5). Con la llegada de CMMI V1.1 se mantienen los cinco niveles de madurez establecidos en el marco anterior, pero cambia la definición del nivel 4 cambia de “Gestionado” a “Gestionado Cuantitativamente” y el nivel 5 pasa a ser “Optimizando” en lugar de “Optimizado”, manteniéndose en las versiones V1.2 y V1.3. En la versión V2.0, evolucionan en su definición para enfatizar el rendimiento del negocio y los niveles de capacidad se aplican a “Áreas de Práctica”, en lugar de a “Áreas de Proceso”. La versión V3.0 introduce en los niveles de madurez el concepto de transformación continua y en la CMMI AIM, definen niveles para el dominio de la IA.

Respecto a la evolución de los niveles de capacidad, se consolidan en la versión V1.1. Esta versión describe 6 niveles (0-5), orientados a evaluar áreas de proceso individuales. La versión V1.2, mantiene el número de niveles, enfocándolos al refinamiento de prácticas. La versión V1.3 orienta mantiene los seis niveles orientándolos hacia la evaluación de las “áreas de proceso”. La versión V2.0, cambia su ámbito de aplicación hacia las “áreas de práctica”, alineando su nomenclatura con la de los niveles de madurez y modificando el nombre del nivel 1 de Inicial a Realizado., manteniéndose en la versión V3.0, aunque se enfocan en medir el rendimiento y la capacidad de innovar. Con la llegada de la versión AIM, define niveles de capacidad para gobernar y maximizar el valor de la IA.

La Tabla 2 resume la evolución de los conceptos a lo largo de la vida del modelo:

Modelo	Año	Enfoque	Niveles de Madurez (Macro)	Niveles de Capacidad (Micro)
SW-CMM	1993	Solo desarrollo de software	5 niveles	No definía este concepto. Solo usaba representación por etapas.

CMMI V1.1	2002	Software, sistemas.	5 niveles:	Sí, 6 niveles (0-5). Evaluación de áreas de proceso individuales.
CMMI V1.2	2006	Software, sistemas,	5 niveles (mismos V1.1)	Sí, 6 niveles (0-5). Refinamiento de prácticas.
CMMI V1.3	2010	Software, sistemas,	5 niveles, renombra ML4 y ML5	Sí, pero simplificados a 4 niveles (0-3) Evaluación de Áreas de Proceso
CMMI V2.0	2018	Unifica constelaciones (DEV, SVC, ACQ) en un solo modelo. Introduce las Vistas	6 niveles (Introduce el nivel 0-Incompleto)	Cambia el ámbito de aplicación a áreas de práctica. Alinea la nomenclatura con los ML Cambia el nombre del nivel 1 de "Inicial" a "Realizado"
CMMI V3.0	2023	Amplía a 8 dominios: desarrollo, servicios, seguridad, datos, personas	5 niveles, pero con enfoque en transformación continua. Sigue los mismos nombres.	Refuerza la medición del desempeño y la capacidad de innovar.
CMMI AIM	2026	Madurez en Inteligencia Artificial (IA)	Modelo de madurez específico para IA, con progresión estructurada.	Define CL para gobernar y maximizar el valor de la IA.

TABLA 2. MATRIZ NIVELES CAPACIDAD VS NIVELES DE MADUREZ

3.3. CMMI V2.0: Rendimiento, valor de negocio y agilidad

La transición de CMMI V1.3 a CMMI V2.0, supone un cambio de paradigma en los modelos de mejora por procesos. A diferencia de sus predecesores, abandona el enfoque normativo y se orienta hacia la mejora continua del rendimiento operativo y la obtención de resultados tangibles (ISACA, 2021b), con un enfoque más adaptable y dinámico. Presenta una arquitectura escalable que se integra con

metodologías ágiles como Scrum y orienta los procesos hacia la mejora del balance de la cuenta de resultados del negocio. (CMMI Institute, s.f.-b, 2019)

3.3.1. Arquitectura de Componentes

El modelo CMMI V2.0 se organiza en cuatro componentes niveles jerárquicos. Esta para facilitar su adopción y orientar los procesos hacia el rendimiento del negocio (CMMI Institute, 2019). La Tabla 3 describe estos componentes, desde el nivel con mayor agregación (Categorías) hasta el más desagregado (Prácticas)."

Nivel Jerárquico	Definición	Propósito	Detalle
1 Categorías (<i>Categories</i>)	Organizan las "Áreas de Capacidad"	Abordar los problemas comunes en la producción.	4 categorías: • "Doing"(Realizar) • "Managing"(Gestión) • "Enabling"(Habilitar) • "Improving"(Mejora)
2 Áreas de Capacidad (<i>Capability Area</i>)	Agrupar conjuntos de "Áreas de Práctica" relacionadas.	Representar habilidades y para un correcto desempeño.	10 Áreas de Capacidad en total.
3 Áreas de Práctica (<i>Practice Areas- PAs</i>)	Colección de prácticas similares que, ejecutadas junta satisfacen el CA	Cada PA se enfoca en una disciplina concreta	25 PAs en total, 18 core (obligatorias para cualquier certificación).
4 Prácticas(<i>Practices</i>)	Acciones específicas a implementar para cumplir con el objetivo de un PA.	Proporcionar los pasos concretos a seguir.	229 prácticas distribuidas en las 25 PAs.

TABLA 3. JERARQUÍA ESTRUCTURAL CMMI V2.0. ADAPTADA DE (CMMI INSTITUTE, 2019)

3.3.2. Niveles de Madurez vs Niveles de Capacidad en CMMI V2.0

CMMI V2.0, unifica los enfoques de madurez y capacidad en un marco flexible, con el objeto de alinear los procesos al rendimiento del negocio en proyectos de

desarrollo tecnológico. Cuando la evaluación se realiza de forma aislada sobre una PA concreta, como (RDM) objeto de este TFM, se evalúa el nivel de capacidad del proceso conforme a los criterios que se expresan en la Tabla 4:

Nivel de Capacidad	Denominación Oficial	Enfoque de Prácticas (Approach)	Impacto en los Activos (Assets)	Rendimiento del Proceso
CL 0	Incompleto (<i>Incomplete</i>)	Ejecución parcial o nula que incumple el propósito base de la AP.	Ausente: Inexistencia de soporte metodológico, documental o normativo estable.	Inconsistente y reactiva: Resultados variables e impredecibles.
CL 1	Inicial (<i>Performed</i>)	Cobertura inicial de la PA destinada a la resolución de urgencias.	Mínimo: Evidencias limitadas al ciclo de vida inmediato de la entrega.	Reactivo: Sin visión de proceso
CL 2	Gestionado (<i>Managed</i>)	Satisfacción íntegra de las prácticas de la AP	Independiente: Activos metodológicos delimitados y controlados a nivel de proyecto.	Controlado: Definición, seguimiento y monitorización activa de objetivos de proyecto.
CL 3	Definido (<i>Defined</i>)	Estandarización organizativa global.	Institucionalizado: Los procesos alimentan de manera recurrente el repositorio corporativo.	Proactivo: Alineación entre la operativa de proyecto y la estrategia de negocio.

TABLA 4. NIVELES DE CAPACIDAD ADAPTADOS CMMI V2.0

En escenarios que requieren alta madurez analítica, la escala de capacidad puede incorporar rangos que utilizan técnicas de control estadísticos y técnicas cuantitativas para controlar y comprender el proceso actual (CL4) y para optimizar los procesos de forma continua (CL5).(Balla et al., 2020)

Cuando el alcance de la evaluación se extiende a toda la organización, ésta se evalúa conforme a los criterios de niveles de madurez que se expresan en la Tabla 5:

Nivel	Nombre	Nivel de desempeño	Nivel de estandarizacion
0	Incompleto (Incomplete)	Es posible que el trabajo no se complete.	No hay procesos establecidos
1	Inicial (Initial)	El trabajo se completa, con posibles retrasos y sobrecostos.	Procesos impredecibles y reactivos.
2	Gestionado (Managed)	Los proyectos se miden, se controlan y se analizan las desviaciones	Cada proyecto se gestiona de forma aislada.
3	Definido (Defined)	Los procesos se gestionan de forma proactiva.	La organización establece estándares de trabajo.
4	Gestionado Cuantitativamente (<i>Quantitatively Managed</i>)	La organización está orientada a datos.	Los objetivos de mejora del rendimiento son cuantitativos y predecibles.
5	En optimización (Optimizing)	La organización es estable y flexible.	Se enfoca en la mejora continua.

TABLA 5.NIVELES DE MADUREZ CMMI ADAPTADO V2.0

3.3.3. Transformación Estructural De Constelaciones a Vistas.

La versión V2.0 rompe con la dicotomía entre “Prácticas Específicas” (SP) y “Prácticas Genéricas” (GP) que caracteriza la versión V1.3, y las unifica en un concepto único de “Prácticas” (*Practices*). Estas se organizan por niveles de evolución de 1 a 5, y se agrupan dentro de las “Áreas de Práctica” (PAs) (*Practice Areas*). Este cambio estructural supone una de las transformaciones más disruptivas del modelo. La versión V1.3 presentaba una estructura rígida organizada en tres constelaciones independientes orientadas a diferentes modelos de negocio: Desarrollo (CMMI-DEV), Servicios (CMMI-SVC) y Adquisiciones (CMMI-ACQ). Cada constelación operaba sobre un núcleo común de 16 áreas de proceso básicas con sus propias “Metas específicas” (SG) y “Prácticas Específicas” (SP), que se repetían para cada área de procesos, y aunque resultaban redundantes para la

organización, había que cumplirlas y verificarlas de cara a una evaluación. La Ilustración 2 presenta la arquitectura del modelo en su versión V1.3.

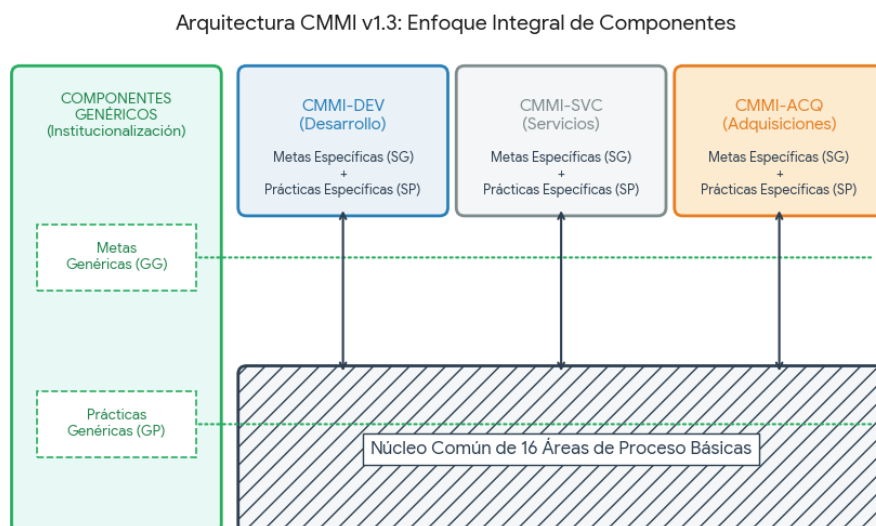


ILUSTRACIÓN 2. ARQUITECTURA INTEGRAL DE COMPONENTES EN CMMI V1.3. ELABORACIÓN PROPIA.

La versión V2.0 corrige esta redundancia, elimina las constelaciones y las integra en un único core. El acceso a las áreas de práctica (PA) se gestiona mediante vistas flexibles, según el modelo de negocio. Las prácticas genéricas se unifican dentro de la categoría de “Sustaining”. Las metas genéricas son reemplazadas por dos áreas de prácticas:

- **Gobernanza (GOV):** Donde se define el rol de la alta dirección en la estrategia, asignación de recursos y definición de responsabilidades.
- **Infraestructura de Implementación (II):** Para asegurar la formalización de los procesos (documentados, seguidos, dotados de recursos y sujetos a mejora continua).

Esto permite que la gobernanza corporativa solo se audite una vez a nivel de organización.

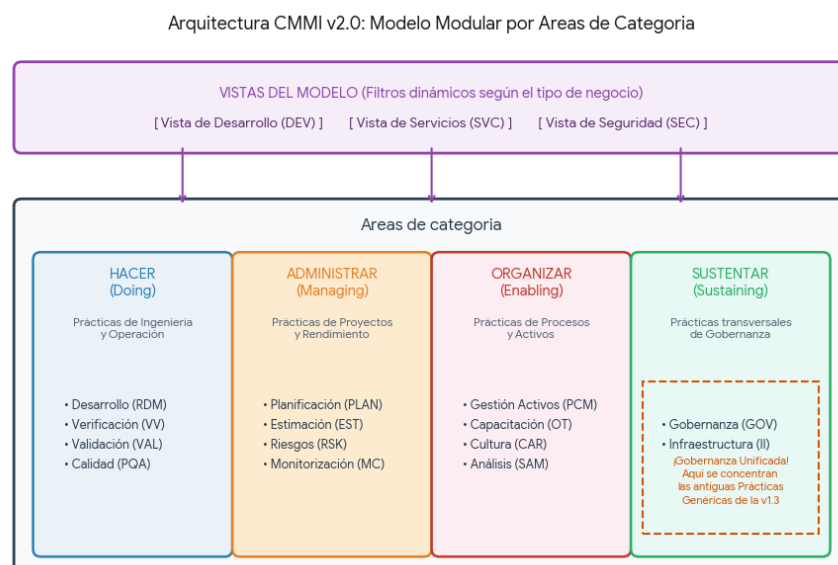


ILUSTRACIÓN 3. ARQUITECTURA INTEGRAL MODULAR EN CMMI V2.0. ELABORACIÓN PROPIA

3.3.4. Integración de prácticas de seguridad en su estructura principal.

En la versión 1.3, la seguridad se gestiona como un elemento externo, en la versión 2.0 se convierte en un dominio específico dentro del modelo, a través de áreas de prácticas específicas. (Niazi et al., 2020; Wood & Vickers, 2018)

3.4. Implicaciones para la Mejora de Procesos

3.4.1. Reorientación estratégica del modelo de la institucionalización al rendimiento:

La versión 2.0, reorienta la estrategia del modelo, centrándose en la mejora de las métricas del rendimiento del negocio. No basta sólo con demostrar la evidencia del cumplimiento de los procesos exige, mejora en las métricas de rendimiento.

Este enfoque en el negocio supone:

- Una adecuación del lenguaje al entorno empresarial y una reducción de la burocracia.
- Una estructura más flexible y escalable, que facilita la integración con metodologías ágiles como Scrum y una adaptación más flexible a las necesidades específicas de cada organización

- La vinculación de las operaciones con indicadores estratégicos de la organización en todos los niveles de madurez (ISACA, 2021a) y mejorar Por otro el método de evaluación para disminuir costes, tiempo y esfuerzo.

La Tabla 6 presenta cómo varía el impacto económico y el retorno de inversión (ROI) entre la versión 2.0, frente a su predecesora:

Criterio de Comparación	Enfoque de ROI en CMMI V1.3	Enfoque de ROI en CMMI V2.0
Ubicación del ROI en el Modelo	Implícito y externo. Se asumía al lograr la certificación de un Nivel de Madurez completo.	Explícito e interno. Cada (PA) y nivel incluye una declaración de valor (Value Statement) que justifica su ROI financiero.
Métrica Principal de Éxito	Cumplimiento normativo (<i>Compliance</i>). Medido por "evidencia objetiva" generada (documentos, plantillas, firmas)	Rendimiento de Negocio (<i>Performance</i>). Medido por KPIs del negocio (costes, velocidad, calidad)
Momento de Retorno	A largo plazo. Beneficios a 18-24 meses tras obtener el ML4 o 5 (Robinson, 2026)	A corto y medio plazo. El ROI se puede calcular de inmediato al implementar prácticas específicas.
Cálculo de Pérdida por Desviaciones	Difícil de aislar. El coste de no calidad se analiza en las áreas de ML	Directo Existe causa-efecto al estar el valor ligado a la práctica.
Rol del Negocio en la Auditoría	El auditor (<i>Lead Appraiser</i>) busca la existencia del proceso. El ROI real no afectaba la nota de la evaluación formal (SCAMPI).	El evaluador busca la mejora en la métrica de rendimiento. La práctica puede considerarse rechazada si su métrica no mejora
Esfuerzo vs. Retorno Financiero	Bajo retorno inicial. Alto coste por burocracia	Alto retorno inicial. (20%-30% de mejora en productividad) (Robinson, 2026)

TABLA 6. CRITERIO DE COMPARACIÓN ENFOQUE DE ROI EN CMMI V1.3 vs CMMI V2.0

CMMI V2.0 vincula directamente los objetivos de negocio con las operaciones, centrándose en la mejora de métricas de factores clave (FunSoft2020, 2020), y

formaliza esta gestión a través del área de proceso "Managing Performance and Measurement" (MPM), mediante un flujo de trabajo predeterminado:

1. Establecer objetivo de negocio, medición y rendimiento.
2. Definir métricas con definiciones operativas claras para medir el progreso.
3. Analizar datos para entender el rendimiento y tomar acciones correctivas.
4. Comunicar resultados a los interesados.
5. Uso de técnicas estadísticas y cuantitativas n niveles de madurez altos (ML4 y ML5), para predecir y optimizar el rendimiento.

Para de sintetizar los cambios estructurales derivados de esta transición arquitectónica y fundamentar su impacto en la hibridación con marcos ágiles, la Tabla 7 contrasta los atributos esenciales de ambas versiones del modelo.

Componente	CMMI v1.3	CMMI v2.0
Organización estructural	Rígida mediante “Constelaciones”	Un solo núcleo consultable mediante <i>Vistas</i> .
Gobernanza de Gestión	Transversal (<i>GPs</i>) repetida en 16 áreas.	Centralizada una vez en la categoría <i>Sustaining</i> .
Burocracia para micropymes	Alta y redundante por la duplicidad de metas.	Baja y optimizada, enfocada en el rendimiento real.

TABLA 7. ATRIBUTOS ESENCIALES ESTRUCTURA V1.3 vs V2.0

3.5. Cambio de modelo a Producto: Implementación y Evaluación:

Con la adquisición del CMMI Institute por parte de ISACA, el modelo se integra en una estructura de producto(ISACA, 2025), que consta de cinco componentes:

- El modelo: Funciona como guía de mejores prácticas para la potenciar el rendimiento empresarial.
- El método de evaluación: Diseñado con criterios específicos para medir el nivel de madurez y la capacidad de la organización.

- Formación y certificación: Consta de un sistema estructurado de cursos de formación para la capacitación profesional y programas de certificación. (CMMI Institute, s.f.-a)
- Herramientas de soporte: Proporcionan recursos de software y plataformas para la gestión del modelo
- Guías de adopción: Orientan a la implementación del modelo a nuevos clientes, y facilitan la transición a empresas certificadas en la versión V 1.3.

A efectos de su aplicación práctica del modelo, esta estructura permite establecer un proceso estandarizado para su implementación. CMMI V2.0, se define como un proceso de ciclo continuo estructurado iterativo en seis pasos (CMMI Institute, 2026b):

Paso 1. Aprender (*Learn*): consiste en entender el modelo CMMI y cómo beneficia a la organización.

Paso 2. Establecer objetivos (*Establish Objectives*): En este punto se Definen las metas de mejora de rendimiento alineadas con el negocio.

Paso 3. Analizar (*Analyze*): Supone, examinar los procesos actuales e identificar brechas de la organización frente al modelo (*gap analysis*).

Paso 4. Desarrollar planes de acción (*Develop Action Plans*): Una vez detectad las brechas, se diseñan las estrategias y planes para mitigar las brechas encontradas.

Paso 5. Desplegar mejoras (*Deploy Improvements*): Supone el despliegue de los planes de acción diseñados de forma práctica las nuevas prácticas en los proyectos.

Paso 6. Evaluar la capacidad y rendimiento (*Assess Capability and Performance*): Introduce la filosofía de mejora continua con un proceso de verificación constante (mediante evaluaciones formales o informales) para validar el progreso.

Flujo de Adopción CMMI V2.0

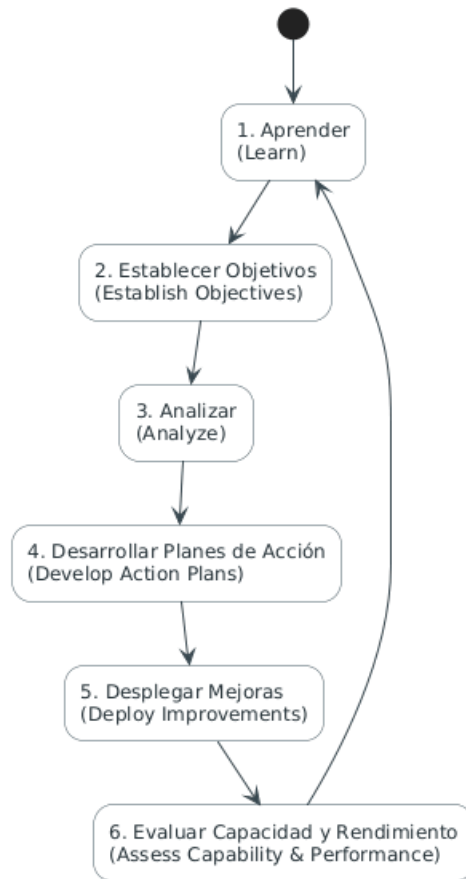


ILUSTRACIÓN 4. DIAGRAMA DE FLUJO DE ADOPCIÓN CMMI V2.0. ELABORACIÓN PROPIA

3.6. CMMI Development V2.0.

El modelo CMMI V2.0 establece los principios y la filosofía general del cambio, pero su aplicación práctica requiere de una concreción sectorial. Para este aterrizaje el modelo ofrece diferentes Vistas. Dado que el objeto de estudio del presente trabajo es el desarrollo de producto de software, la vista que resulta de aplicación es CMMI DEV (Development) V2.0.

La vista CMMI DEV V2.0 constituye la especialización del modelo CMMI V2.0 para el ciclo de vida del desarrollo de productos y software. Esta vista concreta las áreas de práctica, los niveles de capacidad y las métricas para adaptarlas específicamente a la ingeniería de requisitos, diseño, codificación, verificación y validación. A diferencia del esquema monolítico de versiones anteriores, CMMI DEV V2.0 se

organiza en torno a áreas de prácticas, que a su vez contienen prácticas. En el análisis de la literatura existente se presentan diferencias en cuanto al número de prácticas que soporta esta vista del modelo, estas diferencias se explican por la consideración del modelo puro o el modelo expandido, los datos se han extraído del documento oficial del CMMI Institute (CMMI Institute, 2018)

- CMMI-DEV V2.0 Puro: Es el núcleo básico enfocado únicamente en el desarrollo de software y productos (20 Áreas de Práctica). de las cuales 18 son de carácter core y obligatorias para cualquier certificación, y las otras dos, Technical Solution (TS) y Product Integration (PI), son específicas del dominio de desarrollo.
- CMMI V2.0 Extendido: Abarca 25 Áreas de Práctica, las veinte del modelo puro a las que se suman dos áreas :*Servicios*, *Proveedores* y *Data Management* .

Modelo	Categoría	Área Capacidad	de Área de Práctica (PA)
CMMI DEV V2.0			
Puro	4	10	20
Expandido	4	10	25

TABLA 8. RESUMEN CUANTITATIVO DE LA ESTRUCTURA CMMI DEV 2.0

La Ilustración 5, presenta la estructura organizativa del modelo puro:

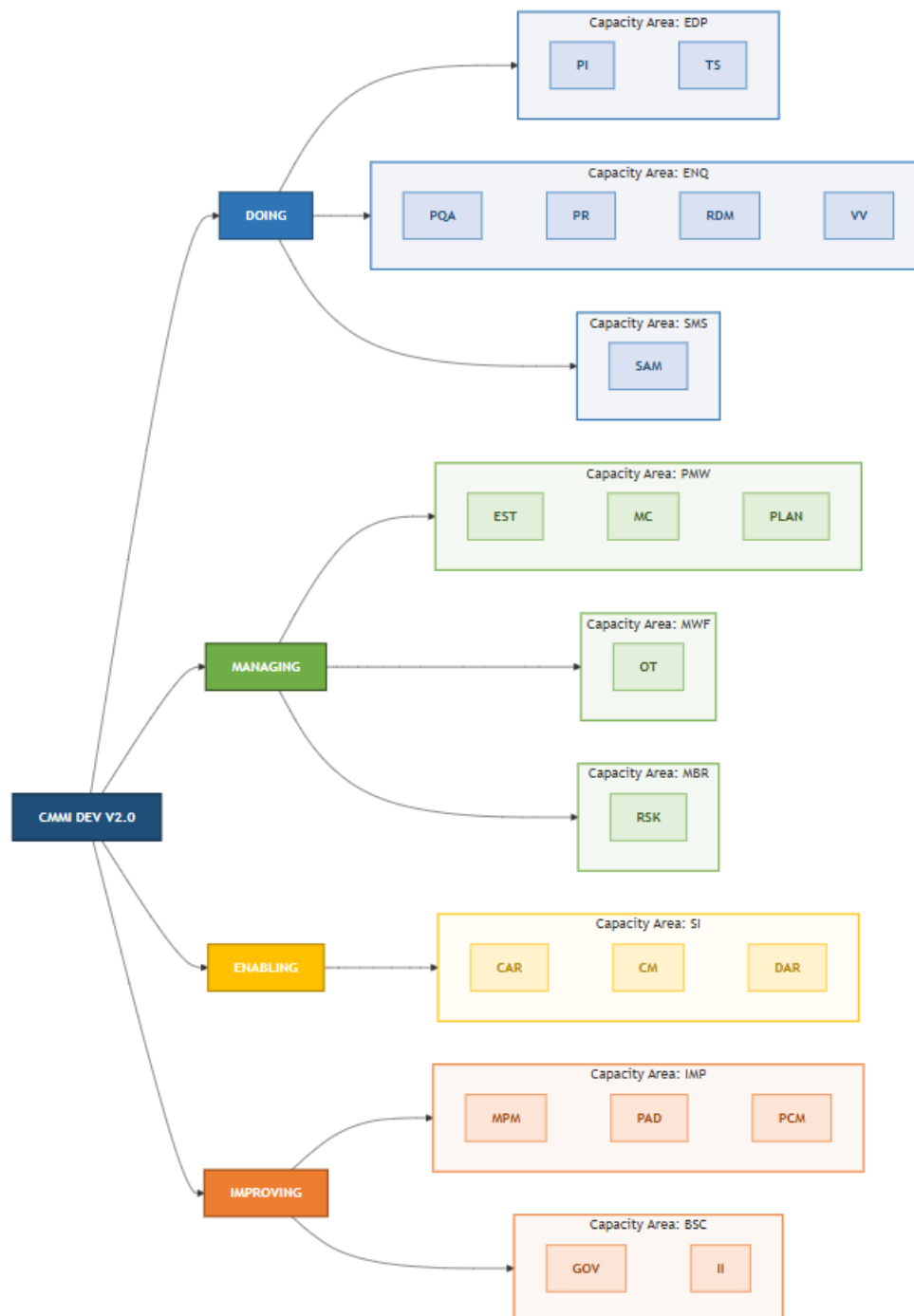


ILUSTRACIÓN 5. ESTRUCTURA CMMI DEV V2.0. ELABORACIÓN PROPIA

De cara a la implementación del modelo, y a su evaluación posterior, es importante conocer la composición de prácticas por nivel. Un nivel de madurez está conseguido si y solo si todas las prácticas correspondientes a dicho nivel están satisfactoriamente implementadas y los niveles anteriores satisfechos, por ello es necesario conocer la granularidad del modelo. En la Tabla 9 se muestra cuantitativamente el desglose por áreas de capacidad y niveles.

El modelo, atendiendo al modularidad buscada, permite la consecución de áreas de practica, para alinearse con las prioridades de la organización. Para ellos es necesario conocer con detalle las diferentes prácticas de un área.

Category	Capacity Area	PA	1	2	3	4	5	Total
DOING	EDP	Engineering and Developing Products	PI	2	12	6		20
DOING	EDP	Engineering and Developing Products	TS	1	3	6		10
DOING	ENQ	Ensuring Quality	PQA	1	4	1		6
DOING	ENQ	Ensuring Quality	PR	1	4	1		6
DOING	ENQ	Ensuring Quality	RDM	1	6	7		14
DOING	ENQ	Ensuring Quality	VV	2	3	2		7
DOING	SMS	Selecting and Managing Suppliers	SAM	3	4	2	1	10
ENABLING	SI	Supporting Implementation	CAR	1	2	5	2	11
ENABLING	SI	Supporting Implementation	CM	1	5			6
ENABLING	SI	Supporting Implementation	DAR	2	5	1		8
IMPROVING	IMP	Improving Performance	MPM	2	6	6	5	22
IMPROVING	IMP	Improving Performance	PAD	1	3	7		11

IMPROVING	IMP	Improving Performance	PCM	3	2	6	1	12
IMPROVING	BSC	Sustaining Habit and Persistence	GOV	1	4	2	1	8
IMPROVING	BSC	Sustaining Habit and Persistence	II	1	2	3		6
MANAGING	PMW	Planning and Managing Work	EST	1	3	2		6
MANAGING	PMW	Planning and Managing Work	MC	2	4	4		10
MANAGING	PMW	Planning and Managing Work	PLAN	2	8	4	1	15
MANAGING	MWF	Managing the Workforce	OT	1	2	6		9
MANAGING	MBR	Managing Business Resilience	RSK	1	2	5		8
Total por nivel				30	84	76	11	4 205

TABLA 9. RESUMEN CUALITATIVOS PRÁCTICAS DESGLOSE POR NIVEL DE MADUREZ CMMI DEV EXTENDIDO 2.0. ADAPTADA DE (CMMI INSTITUTE, 2018)

Por el alto volumen de prácticas, abordar un proyecto de implementación y/o evaluación del modelo en una organización, supone un esfuerzo elevado. Es por tanto necesario establecer un objetivo abordable, como primer requisito, a tomar tras haber formado al equipo en el método. proyecto (véase punto 3.5).

En este sentido este TFM, limita el alcance al desarrollo de una herramienta de auditoría que cumpla los requisitos de un nivel de madurez 2 en el modelo CMMI DEV, y ejecuta su prueba de concepto en el área RDM.

3.6.1. CMMI DEV V2.0 NIVEL DE MADUREZ 2

Aunque el modelo completo de CMMI V2.0 DEV tiene 20 Áreas de Práctica, para obtener el ML2 el ISACA dictamina que hay que sólo hay implementar estas 10 PA: Las demás PAs no se consideran porque sus prácticas están diseñadas para el Nivel 3 o superior.

- Gobernanza y Soporte: GOV, II.
- Gestión y planificación: EST, PLAN, MC.
- Ingeniería básica: RDM.
- Soporte Técnico: CM, PQA, SAM (si aplica).
- Medición: MPM.

Si la implantación del modelo se realiza mediante las áreas de capacidad, un área está satisfecha si sus prácticas individuales lo están. La Ilustración 6 presenta el mapa de áreas de prácticas necesarias para superar el ML2 en CMMI DEV V2.0.

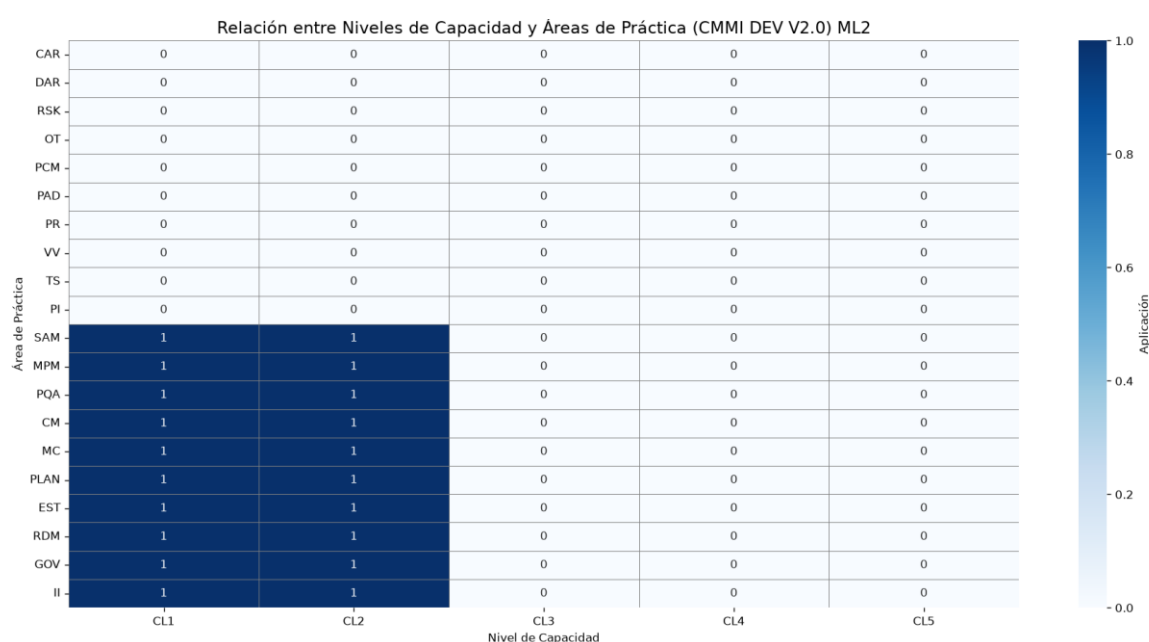


ILUSTRACIÓN 6. PAS DE EVALUACIÓN CMMI DEV 2.0 ML2. ELABORACIÓN PROPIA

Este bloque de procesos se enfoca en estabilizar la gestión operativa de los proyectos. Contempla aspectos críticos como la estimación, planificación y seguimiento del trabajo, así como la gestión de acuerdos con proveedores. Con un volumen de 62 prácticas a cumplir para alcanzar el nivel 2.

Concepto	Dato exacto
Nivel de madurez	Nivel 2 (Gestionado)
Dominios de Práctica (Desarrollo)	10 dominios
Prácticas totales (Desarrollo)	62 prácticas

TABLA 10. RESUMEN CUANTITATIVO DE PRÁCTICAS CMMI DEV V2.0 ML2

Estas PAs se distribuyen en cuatro bloques funcionales, que se detallan a continuación.

Bloque I. Gestión y Planificación

Engloba actividades dedicadas a la planificación y el control operativo de los proyectos, comprende las siguientes PAs:

Practice Area	Prácticas N1	Prácticas N2	Total
EST (Estimación)	1	3	4
PLAN (Planificación)	1	6	7
MC (Monitorización y Control)	1	5	6
SAM (Gestión de Proveedores)	1	7	8

TABLA 11. PAs POR NIVEL CMMI DEV 2.0 ML2. BLOQUE 1

Bloque II. Desarrollo e Ingeniería

Constituye el núcleo técnico del modelo, abarca la gestión de requisitos, el aseguramiento de la calidad del proceso y la gestión de la configuración. Incluye las PAs:

Practice Area	Prácticas N1	Prácticas N2	Total
---------------	--------------	--------------	-------

RDM (Desarrollo y Gestión de Requisitos)	1	5	6
PQA (Aseguramiento de la Calidad del Proceso)	1	4	5
CM (Gestión de la Configuración)	1	5	6

TABLA 12. PAS POR NIVEL CMMI DEV 2.0 ML2. BLOQUE 2

Bloque III. Medición

Agrupar las prácticas relacionadas con la definición, recogida y análisis de métricas para la mejora del rendimiento:

Practice Area	Prácticas N1	Prácticas N2	Total
MPM (Gestión del Rendimiento y Medición)	1	4	5

TABLA 13. PAS POR NIVEL CMMI DEV 2.0 ML2. BLOQUE 3

Bloque IV. Gobernanza y Soporte

Este bloque constituye una de las aportaciones de la versión V2.0, centraliza la gobernanza y la infraestructura de implementación, eliminando la redundancia de las prácticas genéricas de versiones anteriores:

Practice Area	Prácticas N1	Prácticas N2	Total
GOV (Gobernanza)	1	6	7
II (Infraestructura de Implementación)	1	7	8

TABLA 14. PAS POR NIVEL CMMI DEV 2.0 ML2. BLOQUE 4

En su conjunto, CMMI DEV V2.0 ML2 comprende 10 áreas de práctica, que en conjunto despliegan 62 prácticas distribuidas entre los niveles 1 y 2 de capacidad. La Tabla 15 muestra de forma cuantitativa el volumen de prácticas por áreas de prácticas y bloques para superar el ML2:

Bloque	Practice Areas	Prácticas N1	Prácticas N2	Total
I. Gestión y Planificación	EST, PLAN, MC, SAM	4	21	25
II. Desarrollo e Ingeniería	RDM, PQA, CM	3	14	17
III. Medición	MPM	1	4	5
IV. Gobernanza y soporte	GOV, II	2	13	15
TOTALES	10 PAs	10	52	62

TABLA 15. PAs POR NIVEL CMMI DEV 2.0 ML2. RESUMEN

La Tabla 16, resume el número de prácticas y su código de identificación en cada una de las áreas de prácticas necesarias para conseguir el nivel 2 de madurez.

Practice Areas	Prácticas
1. EST (Estimación)	4 prácticas (1.1, 2.1, 2.2, 2.3)
2. PLAN (Planificación)	7 prácticas (1.1, 2.1 a 2.6)
3. MC (Monitorización y Control)	6 prácticas (1.1, 2.1 a 2.5)
4. SAM (Gestión de Proveedores)	8 prácticas (1.1, 2.1 a 2.7)
5. RDM (Desarrollo y Gestión de Requisitos)	6 prácticas (1.1, 2.1 a 2.6)
6. PQA (Aseguramiento de Calidad del Proceso)	5 prácticas (1.1, 2.1 a 2.4)
7. CM (Gestión de la Configuración)	6 prácticas (1.1, 2.1 a 2.5)
8. MPM (Gestión de Rendimiento y Medición)	5 prácticas (1.1, 2.1 a 2.4)
9. GOV (Gobernanza)	7 prácticas (1.1, 2.1 a 2.6)

TABLA 16. LISTADO DE PRÁCTICAS POR PA's CMMI DEV V2.0 ML2

La Ilustración 7, muestra un mapa de calor de las prácticas a superar para la consecución de los diferentes niveles de madurez, en la vista de desarrollo del modelo V2.0-

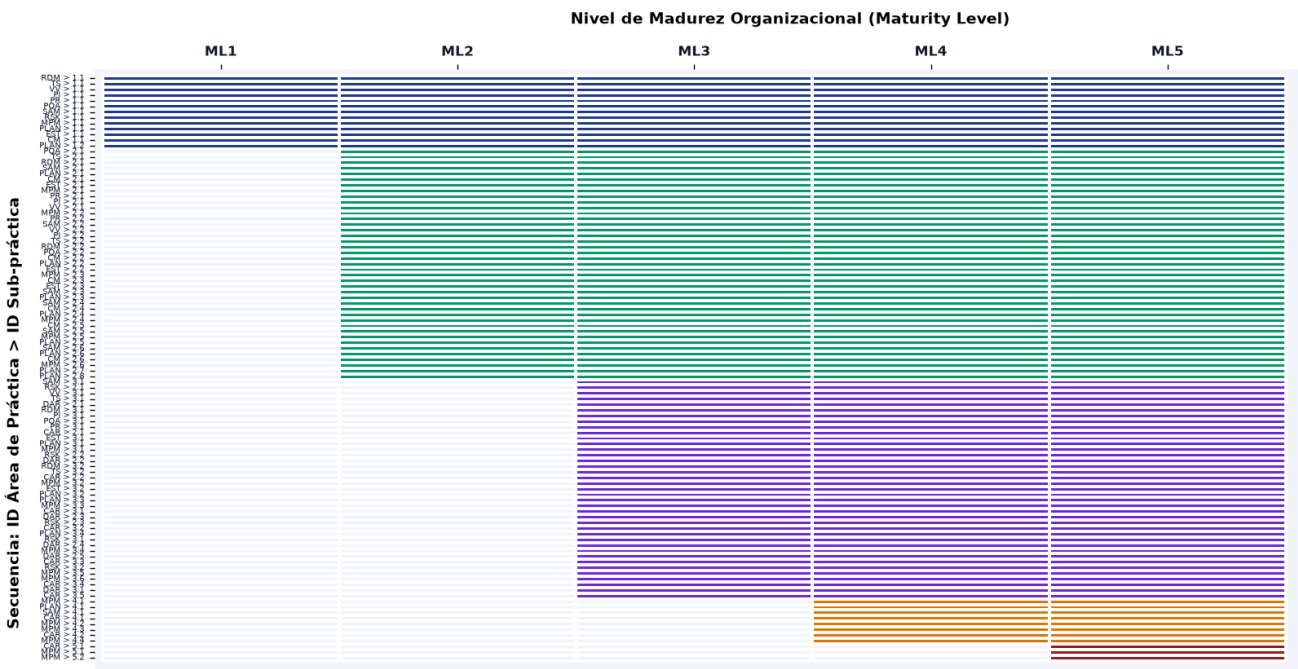


ILUSTRACIÓN 7. MAPA DE CALOR NIVELES MADUREZ CMMI DEV V2.0. ELABORACIÓN PROPIA

CAPITULO 4. Propuesta del modelo conceptual

4. Propuesta del Modelo Conceptual

4.1. Introducción

Este capítulo representa el central de este TFM, ya que en él se despliega todo el diseño metodológico del TFM. El objetivo es materializar la integración entre el modelo CMMI-DEV V2.0, en su Nivel de Madurez 2 ML2, y los flujos operativos del marco ágil Scrum, adaptándolo al contexto específico de una micropyme de consultoría tecnológica. A diferencia del capítulo anterior, que aborda el marco teórico y el estado del arte, este capítulo se centra en la construcción de un modelo híbrido conceptual que sirva de puente entre el rigor normativo de CMMI y la flexibilidad de la operatividad diaria con Scrum.

La propuesta que se desarrolla parte de una premisa básica: la adopción de un modelo CMMI en entornos de pequeña escala, debe plantearse como un medio para la mejora continua del rendimiento. Para lograrlo es preciso diseñar una herramienta que facilite los mecanismos de control, trazabilidad y gobernanza. De este modo la organización puede reducir la incertidumbre en plazos, costes y asegurar la calidad de los entregables.

4.2. Análisis de Contexto.

La organización sobre la que se construye el modelo conceptual es una micropyme de consultoría industrial que, en respuesta a la transformación digital del sector manufacturero, ha orientado su propuesta de valor hacia el desarrollo de software ad-hoc para la optimización de los procesos operativos en planta. Esta evolución ha convertido a la organización en una consultoría tecnológica a pesar de que su núcleo de competencias original proceda de la ingeniería de desarrollo de software.

Esta circunstancia genera una paradoja operativa: por un lado, la organización posee un conocimiento profundo del sector industrial y de los procesos de manufactura de sus clientes, lo que constituye su ventaja competitiva diferencial; por otro lado, carece de procesos maduros de ingeniería de software, lo que se traduce en sobrecostes, retrasos y una calidad variable en los entregables. Esta situación resulta especialmente crítica en un contexto donde los proyectos requieren soluciones a medida, rápidas y fiables.

4.3. Estructura y dinámica de trabajo

La organización objeto de estudio presenta las siguientes características estructurales y operativas:

- **Dimensión del equipo:** La empresa ha creado un área de desarrollo que cuenta con cinco profesionales de forma estable y con una alta dependencia de subcontratación externa para determinadas fases del desarrollo. el equipo está formado por 3 desarrolladores, un analista de negocio y un jefe de departamento. Para el despliegue de Scrum, los roles se han definido así:
 - Product Owner (PO): Función asumida por el analista de negocio.
 - Scrum Master (SM): Rol asumido por el jefe de Departamento y ejerce de jefe de proyecto.
 - Developers: el equipo incluye:
 - Un Full Stack Developer enfocado principalmente en Frontend / interfaz.
 - Un Full Stack Developer enfocado principalmente en Backend / lógica.
 - Un ingeniero con el rol de QA / Tester
- **Modelo de gestión de proyectos:** El equipo opera bajo un marco Scrum con ciclos de Sprint de dos semanas de duración. No obstante, la implementación de Scrum es informal. Los roles no están explícitamente definidos. Las herramientas para la gestión del backlog se han adoptado parcialmente y no existe un sistema consolidado para el seguimiento por métricas para las operaciones. La comunicación con el cliente se realiza siempre a través del Product Owner, que es quien asume la resolución de eventuales bloqueos.
- **Entorno documental:** Mucha de la documentación generada son archivos de Excel o actas de reunión, que se depositan en una carpeta en Drive, sin embargo, este modelo, no es ágil para el seguimiento de los miembros del equipo de proyecto y obliga a tener repositorios dispersos. Otro de los inconvenientes de este tipo de organización documental es la falta de control sobre el código de los productos en su curso intermedio. Sólo se depositan los códigos fuente del

proyecto una vez está cerrado e implementado en cliente. Para corregir estas deficiencias, la empresa ha comenzado un proceso de reestructuración organizativa en la gestión de los proyectos, impulsando el uso de Git, como repositorio principal utilizando archivos puros en formato Markdown (.md) para la redacción del *Product Backlog*, las historias de usuario (*User Stories*), los compromisos del Sprint y la Definición de Hecho (*Definition of Done*, DoD). Este formato facilita la colaboración, pero dificulta la trazabilidad si no se cuenta con un motor semántico.

- **Desafíos identificados:** Entre los problemas recurrentes destacan la ruptura de la trazabilidad bidireccional entre requisitos, código y pruebas. A esto se suma la subjetividad en la validación de entregables, la ausencia de métricas objetivas de progreso y la dificultad para auditar el cumplimiento de estándares de calidad.

4.4. Motivación para la adopción de CMMI

La organización ha identificado la necesidad de adoptar un modelo de mejora de procesos que le permita:

- **Mejorar la predictibilidad:** Reducir la desviación en plazos y costes, hoy problema recurrente en los proyectos de la organización.
- **Incrementar la calidad del producto:** Establecer mecanismos sistemáticos de verificación y validación que reduzcan la dependencia del talento individual.
- **Facilitar la certificación:** Cumplir con los requisitos de calidad exigidos por clientes y licitaciones, que cada vez más, demandan evidencias de procesos maduros.
- **Optimizar la integración con metodologías ágiles:** Conciliar la agilidad de Scrum con la generación de evidencias para CMMI.

La organización se enfrenta a limitaciones presupuestarias que hacen inviable la adopción de soluciones ALM (*Application Lifecycle Management*) empresariales y tampoco puede dedicar personal exclusivo a la gestión de procesos. En este contexto se opta un modelo híbrido ligero y automatizado, que pueda operar nativamente sobre

un entorno Git mediante scripts de verificación en tiempo de ejecución. El modelo debe permitir a la organización ir adaptando sus procesos de forma gradual para consolidar la madurez necesaria para posicionarse como referente en el desarrollo de aplicaciones ad hoc para el sector industrial.

El plazo de adopción, en empresas inmaduras, para adquirir el nivel 2 puede conllevar tiempos cercanos a los 12 meses (Global Quality Services, 2026) . En este proceso las evaluaciones con evidencia de soporte documental suponen un esfuerzo añadido, que las micropymes tienen que reducir.

4.5. Análisis de Brecha (Gap Analysis)

El análisis de brecha es una etapa clave en cualquier proceso de adopción de CMMI, ya que permite identificar las desviaciones entre el estado actual de la organización y los requisitos establecidos por el modelo. La empresa atraviesa un momento de transición, del desarrollo de proyectos industriales a proyectos de desarrollo de software, sus procesos en el ámbito de desarrollo del software dependen más del conjunto de los esfuerzos individuales, que de una metodología estructurada.

Para elaborar el Gap Analysis, se ha seguido la metodología de implantación definida por el CMMI Institute (véase Ilustración 4). Se diseñó un modelo de preevaluación para su uso interno, recogido en el Anexo I. Este análisis se sitúa en la fase de diagnóstico inicial, permitiendo establecer una línea base respecto al su Nivel de Madurez 2,

4.5.1. Metodología de evaluación

El Gap análisis se ha elaborado a partir de una evaluación previa (*Evaluation Appraisal*) mediante diagnóstico interno no oficial en CMMI V2.0. Para ello se ha preparó un formulario para la evaluación de las áreas prácticas extraídas del (*467628891-Booklet-CMMI-V2-0-Model-At-A-Glance-Digital-Spanish-2019-06-28-pdf*, s.f.)

El modelo oficial de evaluación está protegido bajo licencia, y no puede utilizarse. Por este motivo, se diseñó un instrumento de evaluación atendiendo a las directrices del ISACA (Aymerich Fuentes & Jenkins, 2022)

- Un nivel de madurez se considera satisfecho cuando todas las PA de ese nivel, y las de los niveles anteriores, están satisfechas.

- Para que una PA esté satisfecha todas sus prácticas deben estar completa o parcialmente satisfecha. Si alguna no está satisfecha no se considera satisfecha
- Una práctica está satisfecha cuando existe evidencia que lo acredite
- Aun existiendo evidencia, si la métrica de rendimiento no mejora, la práctica puede declararse no satisfecha.

Para cumplir con estas directrices, el modelo de evaluación consta de dos dimensiones complementarias, estructuradas según las 4 categorías del modelo CMMI DEV V2.0. (CMMI Institute, 2019; Valera & Solutions, 2025; van Veenendaal, 2026; Veenendaal et al., 2023):

- Entrevistas con el equipo operativo: El Scrum Master realiza sesiones de diagnóstico técnico con el Product Owner y el equipo de desarrollo. En ellas se da respuesta a un formulario que recorre todas las prácticas del nivel de madurez 2 para CMMI DEV V2.0. y se identifican las evidencias que respaldan las respuestas del formulario
- Evaluación de evidencias documentales: Se eligen dos proyectos y se analizan las evidencias documentales que se han visto durante las entrevistas y verificando su persistencia y trazabilidad.
- Consolidación: Cada práctica es evaluada conforme a los criterios que establece ISACA.

Resultados del diagnóstico inicial

La evaluación responde a una pre evaluación interna, que se realiza, por el Scrum Masterr, sobre las prácticas indicadas por ISACA para evaluar la madurez de la organización de Nivel 2. Para una empresa de tamaño reducido el esfuerzo burocrático es importante, la mayor carga de esfuerzo ha supuesto la búsqueda de la evidencia documental. Los criterios de evaluación aplicados son los que establece la ISACA(Valera & Solutions, 2025; van Veenendaal, 2026; Veenendaal et al., 2023)

Las prácticas son evaluadas de forma cualitativa conforme a los criterios que establece ISACA establece: Totalmente implementada (FI), Mayoritariamente implementada (LI), Parcialmente implementada (PI) y no implementada (NI) atendiendo a las evidencias que se identifican. La clasificación de los criterios se indica en la tabla 17:

Calificación	Significado	Criterio de Aplicación
FI	La práctica se implementa siempre de forma completa y ordenada.	Las evidencias existen en todos los proyectos
LI	La práctica se lleva a cabo en lo más importante, pero faltan algunos detalles en los documentos	Se cumple el proceso de forma generalizada, pero se detectan mejoras que no impactan en el resultado
PI	La práctica se lleva a cabo de forma parcial o desordenada con importantes fallos en su aplicación	El proceso se cumple en algunos proyectos, pero no en otros, o falta algún paso importante.
NI	La práctica no se lleva a cabo o no hay ninguna prueba que acredite su cumplimiento	No existe evidencia documental o es muy incompleta y no tiene valor.

TABLA 17. CRITERIOS DE CALIFICACIÓN ADAPTADOS DE CMMI V2.0.

La tabla 18 resume los resultados por categoría y área de práctica, los criterios de evaluación son:

- Una PA adquiere la condición de SATISFIED, todas sus prácticas tienen que estar valoradas como FI o LI.
- Una Categoría está satisfecha cuando todas sus PA deben tener condición de SATISFIED.
- Para que el Nivel de madurez esté adquirido todas sus categorías están SATISFIED.

CATEGORIA/PA	ESTADO
DOING	NOT SATISFIED
SAM	NOT SATISFIED
RDM	SATISFIED
PQA	NOT SATISFIED
MANAGING	SATISFIED
EST	SATISFIED
PLAN	SATISFIED
MC	SATISFIED
ENABLING	NOT SATISFIED
CM	NOT SATISFIED
IMPROVING	NOT SATISFIED
MPM	NOT SATISFIED
GOV	NOT SATISFIED
II	NOT SATISFIED
Resultado Preevaluacion CMMI DEV 2.0 ML2	NOT SATISFIED

TABLA 18.RESUMEN RESULTADO PRE EVALUACIÓN CMMIDEV 2.0 ML2

La organización no alcanza el Nivel de Madurez 2. Únicamente la categoría MANAGING está SATISFIED. Las categorías DOING, ENABLING e IMPROVING presentan brechas significativas que impedirían la certificación.

La tabla 19, presenta los resultados de la auditoría, por práctica. Los criterios de evaluación aplicados son los que establece la ISACA(Valera & Solutions, 2025; van Veenendaal, 2026; Veenendaal et al., 2023)

DOING		MANAGING		ENABLING		IMPROVING	
ID Práctica	Estado	ID Práctica	Estado	ID Práctica	Estado	ID Práctica	Estado
SAM 1.1	FI	EST 1.1	FI	CM 1.1	LI	MPM 1.1	FI
SAM 2.1	FI	EST 2.1	FI	CM 2.1	LI	MPM 1.2	NI
SAM 2.2	PI	EST 2.2	FI	CM 2.2	NI	MPM 2.1	FI
SAM 2.3	FI	EST 2.3	FI	CM 2.3	PI	MPM 2.2	PI
SAM 2.4	FI	EST	SATISFIED	CM 2.4	FI	MPM 2.3	PI
SAM 2.5	FI	PLAN 1.1	FI	CM 2.5	PI	MPM 2.4	NI
SAM 2.6	PI	PLAN 1.2	FI	CM	NOT SATISFIED	MPM 2.5	NI
SAM 2.7	FI	PLAN 2.1	FI	Supporting Implementation	NOT SATISFIED	MPM 2.6	NI
SAM	NOT SATISFIED	PLAN 2.2	FI	ENABLING	NOT SATISFIED	MPM	NOT SATISFIED
Selecting and Managing Suppliers	NOT SATISFIED	PLAN 2.3	LI			Improving Performance	NOT SATISFIED
RDM 1.1	FI	PLAN 2.4	FI			GOV 1.1	FI
RDM 2.1	FI	PLAN 2.5	FI			GOV 2.1	FI
RDM 2.2	FI	PLAN 2.6	FI			GOV 2.2	LI
RDM 2.3	FI	PLAN 2.7	FI			GOV 2.3	PI
RDM 2.4	LI	PLAN 2.8	FI			GOV 2.4	NI
RDM 2.5	FI	PLAN	SATISFIED			GOV 2.5	NI
RDM 2.6	FI	MC 1.1	FI			GOV 2.6	NI
RDM	SATISFIED	MC 1.2	FI			GOV	NOT SATISFIED
PQA 1.1	FI	MC 2.1	FI			II 1.1	NI
PQA 2.1	NI	MC 2.2	FI			II 2.1	FI
PQA 2.2	NI	MC 2.3	FI			II 2.2	FI
PQA 2.3	NI	MC 2.4	FI			II 2.3	LI
PQA 2.4	FI	MC 2.5	FI			II 2.4	NI
PQA	NOT SATISFIED	MC	SATISFIED			II 2.5	PI
Ensuring Quality	NOT SATISFIED	Planning & Managing Work	SATISFIED			II 2.6	PI
DOING	NOT SATISFIED	Managing	SATISFIED			II 2.7	FI
						II	NOT SATISFIED
						Sustaining Habit and Persistence	NOT SATISFIED
						IMPROVING	NOT SATISFIED

TABLA 19.RESULTADOS AUDITORIA CMMI DEV V2.0 ML2

Análisis detallado por categoría

A continuación, se analiza cada categoría del modelo, desglosando el estado de cada práctica específica, las brechas detectadas y el contraste entre las respuestas del entrevistado y las evidencias presentadas.

• MANAGING

La organización presenta un alto grado de madurez en los procesos de gestión de proyectos. Todas las áreas de práctica de esta categoría se encuentran en estado SATISFIED. Supone una fortaleza donde apoyar el despliegue del modelo

Área de Práctica: EST (Estimación).

Estado: SATISFIED

ID Práctica	Enunciado	Estado	Evidencia Presentada
EST 1.1	Desarrollar estimaciones de alto nivel.	FI	Propuesta Técnica inicial con estimaciones macro archivada en Drive.
EST 2.1	Estimar el alcance del trabajo basándose en una WBS.	FI	<i>Product Backlog</i> estructurado en Épicas e Historias de Usuario.

EST 2.2	Estimar esfuerzo, costo y cronograma con datos históricos.	FI	Documento Excel con Story Points y sesión de <i>Planning Poker</i> documentada.
----------------	--	----	---

EST 2.3	Registrar y mantener bases y suposiciones de las estimaciones.	FI	Matriz de Estimación con supuestos técnicos documentados.
----------------	--	----	---

Análisis de la entrevista: El Scrum Master declaró que las estimaciones se realizan mediante una sesión de *Planning Poker* y se transforman en esfuerzo, duración y coste utilizando la velocidad histórica del equipo. Esta respuesta fue consistente con las evidencias documentales presentadas (Excel de estimaciones y Backlog).

No se detectaron brechas.

Área de Práctica: PLAN (Planificación).

Estado: SATISFIED

ID Práctica	Enunciado	Estado	Evidencia Presentada
PLAN 1.1	Desarrollar una lista de tareas para realizar el trabajo.	FI	Excel con tareas iniciales asignadas por rol.
PLAN 1.2	Asignar personas a las tareas.	FI	Tablero de control con asignación por especialidad.
PLAN 2.1	Desarrollar y mantener el enfoque y plan del proyecto.	FI	Plan de Proyecto con justificación del ciclo de vida.
PLAN 2.2	Planificar la asignación de recursos y conocimiento necesario.	FI	Matriz de Habilidades y Plan de Capacitación.

PLAN 2.3	Desarrollar y mantener el presupuesto y cronograma.	PI	Excel Financiero con historial de <i>Sprints</i> (solo actualizado ante cambios de alcance).
PLAN 2.4	Identificar e involucrar a las partes interesadas.	FI	Matriz de Interesados con roles y frecuencias.
PLAN 2.5	Planificar la transición a operaciones y soporte.	FI	<i>Checklist</i> de Despliegue y actas de entrega.
PLAN 2.6	Obtener el compromiso de los responsables.	FI	Matriz de Capacidad (horas disponibles vs. esfuerzo).
PLAN 2.7	Mantener la consistencia entre los elementos del plan.	FI	Plan de Proyecto unificado con matriz de riesgos, plan de pruebas y configuración.
PLAN 2.8	Revisar los planes y obtener compromisos.	FI	Matriz Rol-Entregable revisada en cada Sprint.

Análisis de la entrevista: El Scrum Master declaró que el plan se revisa de forma continua, pero que las actualizaciones financieras y de calendario solo se realizan ante cambios de alcance significativos, no en cada iteración. Esta respuesta es consistente con la evidencia, que muestra que PLAN 2.3 está en PI por falta de actualización periódica.

Brechas detectadas: Mejorar la frecuencia de actualización del presupuesto y cronograma.

Área de Práctica: MC (Monitorización y Control).

Estado: SATISFIED

ID Práctica	Enunciado	Estado	Evidencia Presentada
-------------	-----------	--------	----------------------

MC 1.1	Revisar las tareas completadas y los resultados.	FI	Panel de control de tareas cerradas.
MC 1.2	Identificar y resolver problemas.	FI	Protocolo de caracterización de problemas y Matriz de Issues.
MC 2.1	Monitorear resultados reales contra el plan.	FI	Gráficas comparativas de 'Planificado vs. Real'.
MC 2.2	Monitorear el involucramiento de los <i>stakeholders</i> .	FI	Registro de entregables del cliente con control de retrasos.
MC 2.3	Monitorear la transición a operaciones y soporte.	FI	Matriz de Riesgos activa y Plan de Transición.
MC 2.4	Identificar, analizar y documentar desviaciones.	LI	Matriz de Issues y Acta de Análisis de Causa Raíz documentada.
MC 2.5	Tomar acciones correctivas y seguirlas hasta el cierre.	FI	Tareas etiquetadas como 'Acción Correctiva' en el Backlog.

Análisis de la entrevista: El Scrum Master declaró que el seguimiento se realiza quincenalmente y mensualmente, y que las desviaciones significativas se registran en una Matriz de Issues, analizando la causa raíz mediante la técnica de los 5 Porqués. Las evidencias presentadas respaldan estas afirmaciones, aunque MC 2.4 está en **LI** porque el análisis de causa raíz no siempre queda documentado de forma estandarizada.

Brechas detectadas: Formalizar el registro del análisis de causa raíz para todas las desviaciones significativas.

La categoría **MANAGING** está completamente implementada (**SATISFIED**). La organización demuestra una sólida capacidad para estimar, planificar, monitorizar y controlar proyectos. Solo se identifican dos oportunidades de mejora leves: **PLAN 2.3** (actualización financiera periódica) y **MC 2.4** (formalización del análisis de causa raíz). Supone una fortaleza para abordar las restantes categorías.

- **DOING**

Esta categoría presenta debilidades significativas que impiden la certificación en Nivel 2. Aunque el área de RDM (Gestión de Requisitos) se encuentra en SATISFIED, las áreas de SAM (Gestión de Proveedores) y PQA (Aseguramiento de Calidad) presentan brechas críticas.

Área de Práctica: RDM (Gestión de Requisitos)

Estado: SATISFIED

ID Práctica	Enunciado	Estado	Evidencia Presentada
RDM 1.1	Registrar los requisitos del proyecto.	FI	Backlog inicial en Excel con requisitos numerados.
RDM 2.1	Identificar, analizar y registrar necesidades de <i>stakeholders</i> .	FI	Actas de Reuniones de Requerimientos.
RDM 2.2	Transformar necesidades en requisitos priorizados.	FI	Historias de Usuario con Criterios de Aceptación.
RDM 2.3	Gestionar los cambios en los requisitos.	FI	Historial de versiones del Backlog y minutas de aprobación.
RDM 2.4	Mantener la trazabilidad bidireccional.	PI	Matriz de Trazabilidad de Requisitos (documental).
RDM 2.5	Asegurar consistencia entre planes y requisitos.	FI	Historial de actualizaciones simultáneas.
RDM 2.6	Asegurar consistencia entre actividades y requisitos.	LI	Análisis de impacto cruzado ante cambios, pero sin formalización completa.

Análisis de la entrevista: El Scrum Master declaró que la trazabilidad se mantiene mediante una matriz de requisitos, y que tienen previsto automatizarla en los Commits de Git. Sin embargo, la evidencia presentada muestra que RDM 2.4 está en PI, ya que la trazabilidad documental existe, pero no está automatizada ni vinculada a los repositorios de código y casos de prueba. RDM 2.6 está en LI porque el análisis de impacto cruzado se realiza, pero no siempre queda registrado de forma sistemática.

Brechas detectadas: Completar la automatización de la trazabilidad en Git y formalizar el registro del análisis de impacto cruzado.

Área de Práctica: SAM (Gestión de Proveedores)

Estado: NOT SATISFIED

ID Práctica	Enunciado	Estado	Evidencia Presentada
SAM 1.1	Monitorear las actividades de los proveedores.	FI	Minutas de reuniones de seguimiento y tablero compartido.
SAM 2.1	Desarrollar y mantener criterios de selección.	FI	Plantilla oficial de 'Criterios de Evaluación'.
SAM 2.2	Seleccionar proveedores basados en capacidades.	PI	Cuadro comparativo de ofertas.
SAM 2.3	Establecer acuerdos formales con proveedores.	FI	Contrato de prestación de servicios y SLAs.
SAM 2.4	Gestionar los cambios en los acuerdos.	FI	Control de cambios y cláusulas de modificación.
SAM 2.5	Revisar y aceptar productos del proveedor.	FI	Protocolo de pruebas y Acta de Entrega-Recepción.
SAM 2.6	Monitorear el desempeño del proveedor.	PI	Reportes métricos de cumplimiento mensual.
SAM 2.7	Evaluar la satisfacción final con el proveedor.	FI	Procedimiento documentado: <i>Checklist</i> técnico, encuesta de satisfacción, evaluación de desempeño y acta de cierre.

Análisis de la entrevista: El Scrum Master declaró que existe un procedimiento formal de cierre que incluye un *checklist* de aceptación técnica, una encuesta de satisfacción a los *stakeholders*, una evaluación de desempeño del proveedor y un acta de cierre, archivándose los entregables en el repositorio.

Brechas detectadas:

- **SAM 2.2 (PI):** La selección final prioriza profesionales de confianza histórica, no solo criterios objetivos documentados.
- **SAM 2.6 (PI):** La medición del desempeño es cualitativa/operativa; falta automatizar el tablero de SLAs.

Área de Práctica: PQA (Aseguramiento de Calidad)**Estado:** NOT SATISFIED

ID Práctica	Enunciado	Estado	Evidencia Presentada
PQA 1.1	Identificar y corregir problemas de calidad.	FI	Tablero de control de defectos (Bugs).
PQA 2.1	Desarrollar una estrategia para el aseguramiento de la calidad.	NI	Ausencia de 'Plan de Aseguramiento de Calidad'.
PQA 2.2	Evaluar objetivamente los procesos ejecutados.	NI	No existen auditorías de proceso.
PQA 2.3	Evaluar objetivamente los productos de trabajo seleccionados.	NI	No existe rol de QA que audite externamente.
PQA 2.4	Comunicar resultados y gestionar no conformidades.	FI	Bitácora de Calidad y correos de notificación.

Análisis de la entrevista: El Scrum Master declaró que no existe un Plan de Aseguramiento de Calidad de Procesos, y que la calidad de los productos reside en el área de *testing* y las actas de validación del usuario. Esta respuesta es consistente con la evidencia presentada, que muestra un vacío crítico en las prácticas PQA 2.1, 2.2 y 2.3.

Brechas detectadas:

- **PQA 2.1 (NI):** Ausencia de un documento metodológico independiente denominado 'Plan de Aseguramiento de Calidad'.
- **PQA 2.2 (NI):** No se ejecutan auditorías de proceso estructuradas bajo un calendario o plan de auditoría formal.

- **PQA 2.3 (NI):** Las revisiones documentales se hacen de manera interna/operativa; falta un rol de QA que audite externamente.

La categoría DOING está NOT SATISFIED. Aunque RDM está en SATISFIED, las áreas de SAM y PQA presentan debilidades críticas que impiden la certificación. Especialmente grave es la situación de PQA, cuyas prácticas 2.1, 2.2 y 2.3 se encuentran en NI, lo que impide verificar objetivamente que los procesos y productos de trabajo cumplen con los estándares definidos. Sin PQA, ningún otro proceso tiene credibilidad ante el evaluador.

- **ENABLING (Habilitación)**

Estado: NOT SATISFIED

Esta categoría agrupa la gestión de configuración, que proporciona la infraestructura de soporte para la ejecución de los proyectos.

Área de Práctica: CM (Gestión de Configuración) – Estado: NOT SATISFIED

ID Práctica	Enunciado	Estado	Evidencia Presentada
CM 1.1	Almacenar y controlar elementos de trabajo esenciales.	LI	Repositorio Drive y Git en proyectos nuevos.
CM 2.1	Identificar los elementos de configuración.	LI	Plan de Proyecto (falta lista explícita de CI).
CM 2.2	Establecer y mantener un sistema de control de configuración.	NI	Sin evidencia de sistema centralizado definitivo.
CM 2.3	Crear o liberar líneas base.	PI	Esquema de carpetas en Drive (sin Tags en Git).
CM 2.4	Registrar y rastrear solicitudes de cambio.	FI	Historial de <i>Pull Requests</i> y control de cambios.

CM 2.5	Realizar auditorías de configuración.	PI	Logs de Git (sin reporte formal de revisión).
---------------	---------------------------------------	----	---

Análisis de la entrevista: El Scrum Master declara que el repositorio actual está cambiando de Drive a Git, y que el control se mantiene mediante un flujo de aprobaciones predefinido. Sin embargo, las evidencias muestran que el sistema centralizado definitivo se encuentra en proceso de despliegue y migración (CM 2.2 en NI).

Brechas detectadas:

- **CM 2.2 (NI):** El sistema centralizado definitivo está en proceso de despliegue y migración; no hay evidencia de un sistema operativo.
- **CM 2.3 (PI):** El control por carpetas físicas debe ser sustituido por el etiquetado (Tags) digital en Git.
- **CM 2.5 (PI):** El Proyecto B no realiza auditorías de configuración; los logs automáticos de Git requieren un reporte formal de revisión.

La categoría ENABLING está NOT SATISFIED. La gestión de configuración presenta brechas subsanables en un plazo corto principalmente relacionadas con la migración a Git y la formalización de auditorías. Pero hay que resolver la situación de la práctica CM 2.2

• **IMPROVING (Mejora):**

Estado. NOT SATISFIED

Esta categoría agrupa las prácticas de medición, gobernanza e infraestructura para la mejora continua. Es la categoría más débil, con varias prácticas en **NI**, aunque se han identificado algunas fortalezas puntuales.

Área de Práctica: MPM (Medición y Gestión del Desempeño)

Estado: NOT SATISFIED

ID Práctica	Enunciado	Estado	Evidencia Presentada
-------------	-----------	--------	----------------------

MPM 1.1	Medir el progreso hacia la finalización de tareas.	PI	Historiales de tareas cerradas y métricas operativas consolidadas por iteración.
MPM 1.2	Identificar y abordar problemas de rendimiento.	NI	Sin evidencia de alineación con objetivos de negocio.
MPM 2.1	Especificar objetivos de medición.	FI	Marco de gestión con metas operacionales integradas.
MPM 2.2	Especificar métricas, procedimientos y herramientas.	FI	Diccionario de Métricas con fórmulas matemáticas y fuentes de datos.
MPM 2.3	Obtener y registrar datos de métricas.	PI	Base de datos en Excel con datos reales capturados.
MPM 2.4	Analizar e interpretar los datos medidos.	PI	Informe de análisis de desviaciones y toma de decisiones documentada.
MPM 2.5	Almacenar datos de medición y resultados.	NI	Sin evidencia de almacenamiento estructurado.
MPM 2.6	Tomar acciones para abordar problemas de rendimiento.	NI	Sin evidencias de acciones correctivas reales derivadas de métricas.

Análisis de la entrevista: El Scrum Master declaró que los datos se consolidan por iteración y que se evalúan para identificar desviaciones, pero que no existen mecanismos formales de revisión por parte de la dirección (MPM 2.5 y 2.6 en NI).

Brechas detectadas:

- **MPM 1.1 (PI)** : La recolección no es diaria; los datos se consolidan por iteración, lo que impide una medición continua del progreso.
- **MPM 1.2 (NI)**: Falta alineación explícita entre objetivos de negocio e indicadores técnicos.
- **MPM 2.4 (PI)**: El proceso de interpretación no está automatizado; faltan minutas de acción correctiva.
- **MPM 2.5 (NI)**: Se requiere formalizar el mecanismo de almacenamiento de datos.
- **MPM 2.6 (NI)**: Faltan minutas de acción correctiva reales derivadas de las métricas.

Área de Práctica: GOV (Gobernanza)

Estado: NOT SATISFIED

ID Práctica	Enunciado	Estado	Evidencia Presentada
GOV 1.1	Identificar a la alta dirección encargada de patrocinar.	FI	Política Corporativa de Procesos firmada por Alta Dirección.
GOV 2.1	Definir y comunicar políticas de ejecución.	FI	Directrices de calidad vigentes.
GOV 2.2	Definir el rol y responsabilidades de gobernanza.	FI	Presupuesto aprobado con partidas para herramientas e infraestructura.
GOV 2.3	Proveer recursos, infraestructura y soporte financiero.	PI	Presentaciones ejecutivas y minutas de reuniones.
GOV 2.4	Asegurar que las competencias sean adecuadas.	NI	Ficha de descripción de puesto (genérica, sin evidencia de rendición de cuentas).
GOV 2.5	Revisar periódicamente el desempeño.	NI	Informe mensual de Objetivos de Departamento (sin evidencia específica por proyecto).
GOV 2.6	Identificar y resolver problemas de escalamiento.	FI	Matriz de bloqueos y correos institucionales de resolución.

Análisis de la entrevista: El Scrum Master declaró que la dirección delega la supervisión en la Jefatura de Departamento y que existen reuniones periódicas de dirección. Sin embargo, la evidencia muestra que las revisiones se realizan a nivel macro/departamental, no a nivel de proyecto específico (GOV 2.4 y 2.5 en NI).

Brechas detectadas:

- **GOV 2.4 (NI):** No hay evidencia de que la dirección asegure las competencias del personal mediante evaluación formal.
- **GOV 2.5 (NI):** Las revisiones de desempeño no se realizan a nivel de proyecto, sino a nivel departamental.

- **GOV 2.3(PI):** La evidencia de provisión de recursos es genérica; falta especificidad por proyecto.

Área de Práctica: II (Infraestructura de Implementación)

Estado: NOT SATISFIED

ID Práctica	Enunciado	Estado	Evidencia Presentada
II 1.1	Ejecutar los procesos siguiendo un enfoque documentado.	NI	Guías rápidas (sin procedimiento de control formal).
II 2.1	Desarrollar, mantener y documentar los procesos.	FI	Control de versiones del Plan de Calidad.
II 2.2	Planificar la adopción y despliegue de procesos.	FI	Listas de asistencia a capacitaciones.
II 2.3	Proveer entrenamiento o asegurar habilidades.	LI	Inventario de accesos y auditorías de actividad.
II 2.4	Asegurar que herramientas y recursos estén disponibles.	FI	Inventario de accesos asignados y reportes de auditoría de actividad.
II 2.5	Monitorear y evaluar la efectividad de los procesos.	PI	Registro cualitativo de cuellos de botella.
II 2.6	Identificar oportunidades de mejora.	PI	Formulario Excel de Solicitudes de Mejora con estados de evaluación.
II 2.7	Almacenar lecciones aprendidas.	FI	Matriz o repositorio digital de Lecciones Aprendidas en Drive.

Análisis de la entrevista: El Scrum Master declaró que la evaluación de la efectividad de los procesos se realiza de forma cualitativa y continua. Sin embargo, la evidencia muestra que II 1.1 está en NI (falta un procedimiento estricto de verificación y control).

Brechas detectadas:

- **II 1.1 (NI):** Falta implementar un procedimiento estricto e independiente de verificación y control.

- **II 2.5 (PI):** La evaluación de efectividad es cualitativa; falta un mecanismo métrico estandarizado.
- **II 2.6 (PI):** El Proyecto B no registra activamente propuestas de mejora.

La categoría IMPROVING es la más débil de toda la evaluación, con múltiples prácticas en NI, se recomienda abordar esta categoría en una fase posterior y centrar los esfuerzos inmediatos en las otras tres categorías.

Resumen de Brechas Identificadas

La Tabla 20 clasifica las brechas detectadas según su severidad, utilizando la nomenclatura estándar de CMMI.

Severidad	Práctica Afectada	Descripción de la Brecha	Impacto en la Certificación
CRÍTICA (NI)	PQA 2.1	Ausencia de un Plan de Aseguramiento de Calidad como documento metodológico independiente.	BLOQUEA: sin estrategia de calidad, no hay verificación objetiva.
CRÍTICA (NI)	PQA 2.2	No se ejecutan auditorías de proceso bajo un calendario o plan formal.	BLOQUEA: sin auditorías, no hay evidencia de cumplimiento.
CRÍTICA (NI)	PQA 2.3	Las revisiones documentales son internas; falta un rol de QA que audite externamente.	BLOQUEA: sin auditoría externa, no hay objetividad.
CRÍTICA (NI)	CM 2.2	El sistema centralizado de control de configuración está en migración; no hay evidencia de un sistema operativo.	BLOQUEA: sin sistema operativo, CM no puede estar SATISFIED.
MEDIA (PI)	SAM 2.2	La selección de proveedores prioriza la confianza histórica sobre criterios objetivos documentados.	Dificulta que SAM esté SATISFIED.

MEDIA (PI)	SAM 2.6	La medición del desempeño del proveedor es cualitativa; falta automatizar el tablero de SLAs.	Dificulta que SAM esté SATISFIED.
MEDIA (PI)	CM 2.3	El control de líneas base se realiza por carpetas físicas, no por etiquetado digital en Git.	Dificulta que CM esté SATISFIED.
MEDIA (PI)	CM 2.5	Los logs de Git no se acompañan de un reporte formal de auditoría de configuración.	Dificulta que CM esté SATISFIED.
MEDIA (PI)	PLAN 2.3	El presupuesto y el cronograma solo se actualizan ante cambios de alcance, no en cada iteración.	No bloquea, pero debe corregirse.
MEDIA (PI)	MPM 1.1	La recolección de datos no es diaria; se consolida por iteración.	Afecta a IMPROVING.
MEDIA (PI)	MPM 2.3	La base de datos de métricas está en Excel, sin integración con el sistema de gestión.	Afecta a IMPROVING.

TABLA 20. CLASIFICACIÓN DE BRECHAS POR SEVERIDAD

Consistencia de la información reportada

Para verificar la consistencia de la información reportadas, se realiza un contraste entre las declaraciones del Scrum Master y las evidencias documentales presentadas. La Tabla 21 los puntos más significativos.

Práctica	Declaración del Scrum Master	Evidencia Presentada	Resultado
SAM 2.7	Existe un procedimiento que incluye checklist técnico, encuesta de satisfacción y acta de cierre.	Procedimiento documentado y	Consistente (FI).

		evidencias de aplicación.	
MPM 1.1	Medimos lo que importa al negocio: trazamos cada métrica hasta un objetivo estratégico, lo documentamos en un plan y lo actualizamos cada sprint."	No existe un 'Plan de Medición' formal; los datos se consolidan por iteración.	Inconsistente (PI).
CM 2.2	Estamos en fase de migración hacia Git. El control actual se mantiene mediante un flujo de aprobaciones."	No hay evidencia de un sistema centralizado definitivo.	Parcial (NI).
PQA 2.1	No existe Plan de Aseguramiento de Calidad de Procesos."	Ausencia total de auditorías de proceso.	Consistente (NI).
RDM 2.4	Tenemos una matriz de trazabilidad y prevemos automatizarla."	Matriz documental existente, pero sin automatizar.	Parcia (PI).
GOV 2.6	Registramos el bloqueo en una matriz de bloqueos, lo escalamos formalmente mediante correo."	Matriz de bloqueos y correos institucionales disponibles.	Consistente (FI).

TABLA 21.CONTRASTE ENTRE DECLARACIONES Y EVIDENCIAS

Existen problemas de formalización en varias prácticas. Algunas prácticas se ejecutan de manera operativa, pero no están documentadas ni estandarizadas. A efectos de evaluación, no se considerarían implementadas. Existen oportunidades de mejora.

Conclusiones del Diagnóstico Inicial

El diagnóstico inicial, basado en el modelo de preevaluación que recoge el Anexo1 y en el análisis de evidencias documentales y entrevistas, determina que la organización, no alcanza el nivel de madurez requerido para la certificación CMMI V2.0 Nivel 2, ya que únicamente la categoría MANAGING (Gestión) se encuentra en estado SATISFIED. Esta categoría supone una fortaleza, porque los procesos de estimación, planificación, monitorización y control son sólidos y están respaldados por evidencias consistentes.

Por otro lado, en la categoría DOING, el área de prácticas RDM (Gestión de Requisitos) está en estado SATISFIED, lo que demuestra que la organización sabe gestionar los requisitos desde su captura hasta su implementación.

Si embargo existen brechas críticas, PQA (Aseguramiento de Calidad) está NOT SATISFIED. Las prácticas PQA 2.1, 2.2 y 2.3 están en NI. Esta es la brecha más crítica, ya que, sin auditorías objetivas de procesos y productos, ningún otro proceso tiene credibilidad, en CM (Gestión de Configuración) falta un sistema centralizado de control de cambios, y mejorar el cumplimiento de prácticas como CM 2.3 y 2.5 en PI, que están Parcialmente implementadas. Respecto al área de IMPROVING (Mejora) está NOT SATISFIED, con varias prácticas en NI. Sin embargo, su impacto no es operativo, por lo que puede abordarse en fase posterior. A partir de los resultados del diagnóstico y como resultado de esta evaluación se establece un plan de acción con prioridades, que recoge la tabla 22.

Prioridad	Categoría	Acción
1	DOING	Implementar auditorías objetivas de procesos y productos (PQA 2.1, 2.2, 2.3).
2	ENABLING	Implementar un sistema centralizado de control de configuración (CM 2.2).
3	DOING	Completar la selección y seguimiento de proveedores (SAM 2.2, 2.6).
4	ENABLING	Automatizar líneas base y auditorías de configuración (CM 2.3, 2.5).
5	MANAGING	Mejorar actualización financiera periódica (PLAN 2.3) y formalizar análisis de causa raíz (MC 2.4).
6	IMPROVING	Abordar en fase posterior, aunque también es requisito para ML2

TABLA 22. PLAN DE ACCIÓN PRIORIZACIÓN

Este plan de acción, responde a la solución tradicional de la mejora de procesos, que requiere contratación de servicios especializados de consultoría y la implantación de sistemas ALM (Application Lifecycle Management) de alto coste y la dedicación de personal exclusivo a tareas de aseguramiento de la calidad. Sin embargo, dichas

alternativas resultan inviables para organización, ralentizaría los ciclos de desarrollo y desvirtuaría la esencia ágil que desde el equipo trabaja bajo este modelo de Scrum adaptado.

Por otra parte, la naturaleza de las brechas detectadas apunta a un problema común: la dificultad para generar, mantener y verificar de forma continua las evidencias objetivas que demanda CMMI. La trazabilidad entre requisitos, código y pruebas se rompe porque no existe un mecanismo automático que la actualice; las auditorías de proceso no se realizan porque requerirían un esfuerzo manual periódico; la gestión de la configuración se resiente porque el control de cambios y la identificación de líneas base dependen de la disciplina individual de los desarrolladores. En definitiva, el problema responde a la falta de herramientas ligeras y automatizadas que integren el control normativo en el flujo de trabajo diario, sin imponer sobrecargas operativas.

Es precisamente en este punto donde se justifica la propuesta central de este Trabajo Fin de Máster: el desarrollo de un motor de auditoría automatizada, concebido como un artefacto software ad-hoc que actúa como un "garante de cumplimiento". Este motor, programado en Python y basado en una arquitectura FastAPI con persistencia en SQLite, se integra de forma no intrusiva en el repositorio Git del equipo, escanea los archivos Markdown que contienen el *Product Backlog*, las historias de usuario, los compromisos de Sprint y la Definición de Hecho (DoD), y aplica expresiones regulares para verificar la conformidad con las prácticas de CMMI.

En el diseño de la herramienta se ha empezado por el Área de Práctica RDM (Desarrollo y Gestión de Requisitos) porque es un área dentro de la categoría DOING que está perfectamente satisfecha. A nivel operativo es mucho más adecuado empezar la automatización con procesos estables. No obstante, este motor se desarrolla con una filosofía modular con capacidad de extensión al resto de áreas que permite:

1. **Automatización de la auditoría continua:** el motor permite realizar verificaciones en tiempo de ejecución, cada vez que el equipo integra código o actualiza la documentación, eliminando la necesidad de auditorías retrospectivas manuales y reduciendo el riesgo de error humano.
2. **Trazabilidad bidireccional garantizada:** al establecer relaciones de clave foránea en la base de datos entre historias de usuario, criterios de aceptación,

resultados de pruebas y hallazgos de auditoría, el sistema asegura que cada requisito esté vinculado a su implementación y a su verificación, cumpliendo así con RDM 2.4 y RDM 2.6 de forma automatizada.

3. **Generación objetiva de evidencias:** el motor produce reportes estructurados y métricas de porcentaje de criterios validados, que sirven como evidencias objetivas ante futuras evaluaciones formales, atacando directamente las debilidades de PQA y MPM.
4. **Bajo coste y adaptabilidad:** al estar desarrollado con tecnologías de código abierto y funcionar sobre un repositorio Git estándar, no requiere inversión en herramientas propietarias ni en infraestructura adicional, lo que lo hace asequible y escalable para micropymes.
5. **Integración ágil:** el motor no impone cambios drásticos en la metodología de trabajo; se limita a auditar los artefactos nativos de Scrum, de modo que el equipo mantiene su agilidad mientras recibe retroalimentación inmediata sobre su grado de cumplimiento normativo.

El desarrollo de esta Prueba de Concepto (PoC) supone la solución a un problema real detectado en el diagnóstico. La herramienta no solo permitirá a la organización avanzar hacia la certificación CMMI ML2, sino que también sentará las bases para una cultura de mejora continua basada en datos objetivos y trazables.

4.6. Mapeo metodológico y Matriz de trazabilidad cruzada:

Para asegurar la validez en un entorno híbrido CMMI-DEV v2.0-Scrum, es indispensable establecer la relación entre las prácticas Scrum y el cumplimiento de las prácticas que exige el estándar. Con este propósito Henríquez et al, (Henriquez et al., 2022), realizan un mapeo de correspondencia con artefactos entre artefactos SCRUM y prácticas CMMI V2.0 , la tabla 23 , presenta esta relación , para las prácticas de la vista de desarrollo en el cumplimiento de su madurez ML2.

4.6.1. Análisis de la Matriz de Relación PA CMMI DEV V2.0 ML2-Artefacto Scrum

El eje vertical del gráfico clasifica los bloques funcionales de CMMI (*Doing, Enabling, Improving, Managing*), divididos según los códigos de sus respectivas PAs (RDM, PI,

TS, PR, VV, CM, MPM, RSK, EST, MC y PLAN). Por otro lado, la dimensión horizontal reúne un catálogo de 21 facilitadores y prácticas del ecosistema ágil, abarcando desde eventos presenciales (*Daily Scrum*, *Sprint Planning*) hasta entregables y herramientas automatizadas (el *Product Backlog*, las directrices de la *DoD*, o los procesos de *Continuous Integration/Build*).

Este mapa de correspondencias demuestra que existe integridad en el mapeo de los artefactos Scrum bajo el modelo de funcionamiento CMMI DEV 2.0 ML2 y que facilita:

- **La gestión de requerimientos (RDM):** El desarrollo del software propuesto en esta POC (que cubre las prácticas RDM 1.1 a 2.6) se apoya en la gestión de los proyectos mediante Scrum. Los registros en amarillo demuestran que la conformidad con RDM no depende de un único registro aislado. Es el resultado de la dinámica diaria del equipo: Refinamiento del *Backlog Grooming*, la evolución de las historias de usuario, la negociación de alcance en planificación y la verificación final de requisitos con cliente durante el *Sprint Review*.
- **Aseguramiento y verificación (PR / VV):** Las exigencias de revisión (PR) y verificación (VV) se cubren de forma nativa con herramientas técnicas. En lugar de auditorías manuales, usamos flujos de integración continua (CI), desarrollo guiado por pruebas (TDD) y el cumplimiento estricto de los criterios de calidad definidos en la Definición de Hecho (DoD).
- **Métricas para el control y la estimación (PLAN / EST / MC):** La planificación y el control se asumen por Scrum. Las estimaciones se basan en el consenso del equipo (*Team Estimation Game*), el día a día se gestiona con los acuerdos internos (*Team Agreements*) y el avance se mide visualmente con los diagramas de *Burndown*. Las métricas de control son más realistas que en entornos tradicionales.

	Backlog Grooming	Burndown Charts	Continuous Integration/Build	Daily Scrum	Definition of Done	Epics	Historical velocity	Pair Programming	Product Backlog	Refactoring	Release Planning	Sprint Execution	Sprint Planning	Sprint Review/Demo	Team Agreements	Team Estimation Game	Technical Debt	Test Driven Development	User Stories
Doing																			
Engineering & Developing Products																			
RDM																			
RDM 2.1																			
RDM 2.2																			
RDM 2.3																			
RDM 2.4																			
RDM 2.5																			
RDM 2.6																			
Engineering & Developing Products																			
PI																			
PI 2.1																			
PI 2.2																			
PI 2.3																			
PI 2.4																			
TS																			
TS 2.1																			
Ensuring Quality																			
PR																			
PR 2.1																			
PR 2.3																			
PR 2.4																			
VV																			
VV 1.1																			
VV 1.2																			
VV 2.1																			
VV 2.2																			
VV 2.3																			
Enabling																			
Supporting Implementation																			
CM																			
CM 2.4																			
Improving																			
Managing Business Performance																			
MPM																			
MPM 2.1																			
MPM 2.2																			
MPM 2.3																			
MPM 2.4																			
MPM 2.5																			
Managing																			
Managing Business Resilience																			
RSK																			
RSK 1.1																			
RSK 2.1																			
RSK 2.2																			
Planning & Managing Work																			
EST																			
EST 2.1																			
EST 2.2																			
EST 2.3																			
MC																			
MC 2.1																			
MC 2.2																			
MC 2.4																			
PLAN																			
PLAN 2.1																			
PLAN 2.3																			
PLAN 2.4																			
PLAN 2.6																			
PLAN 2.7																			
PLAN 2.8																			
Total general																			

TABLA 23.RELACIÓN PRÁCTICAS CMMI DEV V2.0 ML2-ARTEFACTOS SCRUM. ADAPTADA
(HENRIQUEZ ET AL., 2021, 2022)

4.7. Roles y responsabilidades en el modelo híbrido

4.7.1. Definición de roles

La Tabla 24 recoge las responsabilidades de cada uno en relación con las prácticas de CMMI.

Rol	Responsabilidad principal	Prácticas asociadas	CMMI
Product Owner	Captura y prioriza requisitos. Aprueba cambios y líneas base.	RDM 1.1, RDM 2.1, RDM 2.2, RDM 2.3	
Scrum Master	Facilita los eventos Scrum. Supervisa el cumplimiento del proceso.	RDM 2.4, MC 2.4, II 2.5	
Developers	Implementan, hacen <i>commits</i> , mantienen la trazabilidad.	RDM 2.5, RDM 2.6, CM 2.4	
Auditor Interno (ATM)	Valida hallazgos, gestiona acciones correctivas, audita la configuración.	PQA 2.2, PQA 2.3, CM 2.5	
Lead Appraiser	Rol externo. Solo interviene en evaluaciones formales.	GOV 2.5, RDM ML2	

TABLA 24. DEFINICIÓN DE ROLES MODELO HÍBRIDO.

4.7.2. Matriz de casos de uso (modelo conceptual)

La Tabla 25 relaciona cada caso de uso con sus actores y la práctica CMMI que cubre. No incluye flujos técnicos ni patrones de búsqueda: esos detalles se recogen en el apartado 5.4.

Código	Nombre	Actores (principal / secundario)	Práctica CMMI
CU-01	Crear e inicializar proyecto	Product Owner / Scrum Master	RDM 1.1
CU-02	Crear y aprobar historias de usuario	Product Owner / Developer	RDM 2.1, RDM 2.2
CU-03	Identificar y gestionar riesgos	Scrum Master / Product Owner	RDM 2.1
CU-04	Planificar sprints y tareas	Developer / Scrum Master	RDM 2.3, RDM 2.4
CU-05	Registrar versión de artefacto	Developer / Scrum Master	RDM 1.1
CU-06	Gestionar y aprobar línea base	Developer / Product Owner	RDM 2.6
CU-07	Ejecutar escenario automatizado	Developer / Scrum Master	RDM 2.6
CU-08	Validar alertas y falsos positivos	Auditor Interno / Developer	RDM 2.6
CU-09	Gestionar hallazgos y acciones	Auditor Interno / Scrum Master	RDM 2.6
CU-10	Ver historial y trazabilidad	Auditor Interno / Lead Appraiser	RDM 2.5

Código	Nombre	Actores (principal / secundario)	Práctica CMMI
CU-11	Firmar y emitir informe	Lead Appraiser / Alta Dirección	RDM ML2
CU-12	Gestionar cambios en configuración	Developer / Product Owner	RDM 1.1
CU-13	Auditar integridad de configuración	Scrum Master/ Auditor Interno	RDM 2.6

TABLA 25. MATRIZ CASOS DE USO MODELO CONCEPTUAL

4.7.3. Matriz RACI

La Tabla 26 resume quién es responsable (R), quién aprueba (A), quién consulta (C) y quién informa (I) en cada fase del ciclo.

Actividad	PO	SM	Dev	ATM	Lead Appraiser
Captura de requisitos	R/A	C	C	I	
Priorización del backlog	R/A	C	C	I	
Sprint <i>Planning</i>	C	R	R	I	
Implementación	I	C	R/A	I	
Auditoría automática	I	C	I	R/A	

Actividad	PO	SM	Dev	ATM	Lead Appraiser
Gestión de hallazgos	C	R	R	A	I
<i>Sprint Review</i>	R	R	R	C	I
Retrospective	C	R/A	R	C	
Evaluación formal	I	C	I	C	R/A

TABLA 26. MATRIZ RACI

CAPITULO 5. Desarrollo técnico y validación

5. Desarrollo Técnico y Validación:

El diseño de la solución propuesta aborda la necesidad de unificar la flexibilidad operativa del marco de trabajo ágil Scrum con las exigencias normativas y de control de procesos que establece el modelo CMMI DEV v2.0 en su Nivel de Madurez 2 (ML2).

5.1. Descripción de alto nivel

La solución que se plantea, es una Prueba de Concepto (PoC) de un sistema modular y extensible. La herramienta demuestra ser viable técnicamente para automatizar la auditoría de procesos CMMI ML2 dentro del ciclo de vida diario de un equipo ágil, sin necesidad de recoger de forma manual de evidencias documentales.

La solución presenta una arquitectura modular estructurada en 3 capas:

Capa 1. Capa de Interfaz y Enrutamiento (WebUI / API), construida sobre FastAPI (`src/dashboard.py` y `app/main.py`). Se encarga de gestionar las peticiones de la web, mostrar los datos en pantalla y leer el archivo `estructura.txt` para pintar las pestañas de las áreas de CMMI de forma dinámica.

Capa 2. Capa de Lógica de Negocio y Procesamiento (Motor de Auditoría) donde residen `motor_auditoria.py` y las reglas de `rdm_scrum.py`. Aquí se cargan las expresiones regulares, se invoca a la librería `GitPython` para clonar el repositorio y se analizan los archivos del proyecto

Capa 3. Capa de Persistencia (Base de Datos):: Implementada con el ORM SQLAlchemy y los modelos de datos en la carpeta `src/db_persistence/` (como `proyecto.py`, `resultado.py`, `user_story.py`) que guardan la información de forma permanente en la base de datos SQLite.

Para lograr modularidad y extensibilidad, se realiza una configuración en tres elementos,: un manifiesto (*estructura.txt*) que define los módulos activos, un conjunto de módulos de reglas (*rdm_scrum.py*) y un motor (*motor_auditoria.py*) que carga el módulo correspondiente según el manifiesto

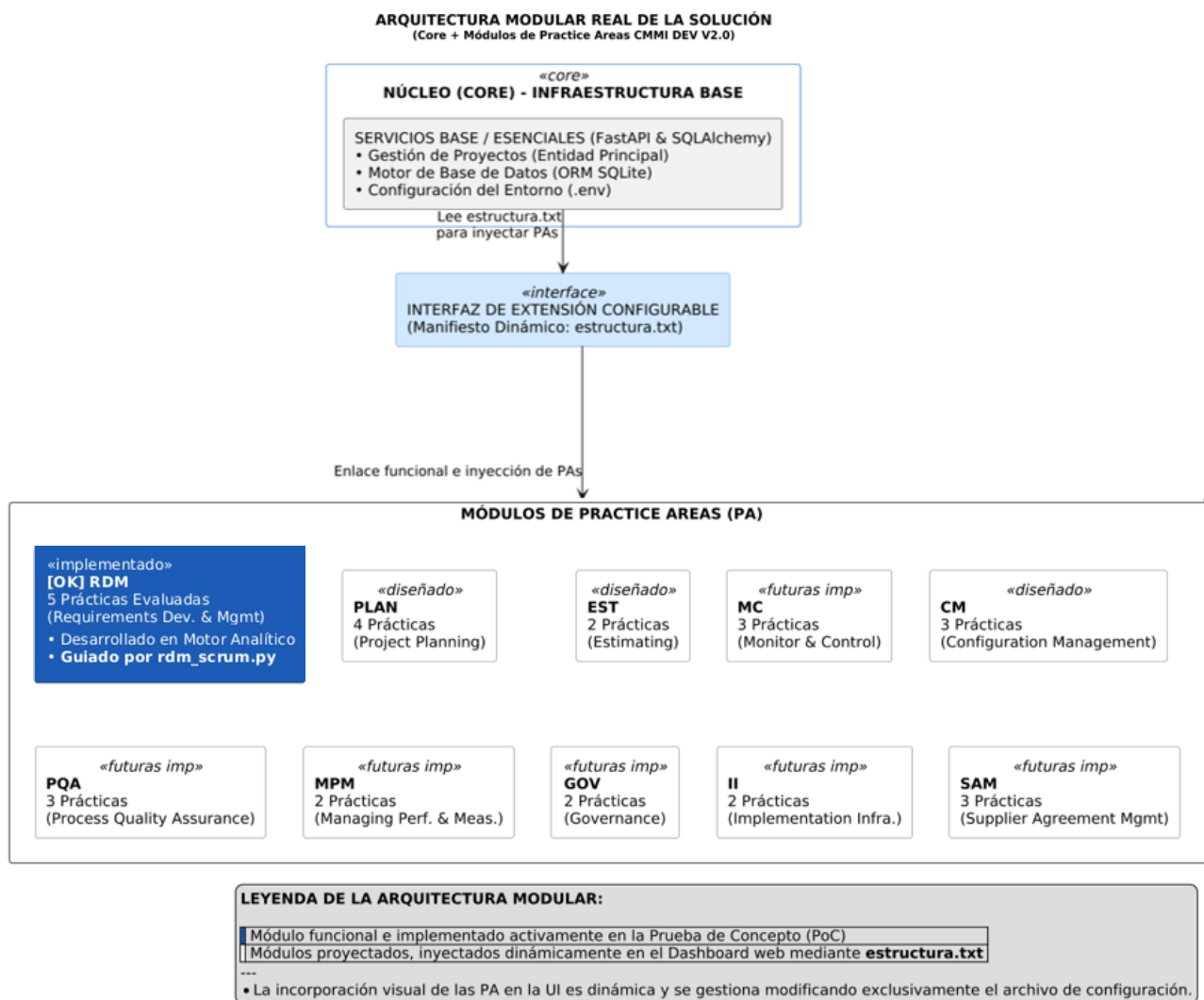


ILUSTRACIÓN 8. ARQUITECTURA LÓGICA MODULAR BASADA EN COMPONENTES

5.2. Stack Tecnológico

La solución se construye sobre el ecosistema Python empleando el framework FastAPI, la persistencia con, SQLite gestionada mediante el ORM SQLAlchemy, lo que evita configurar servidores de bases de datos externos y facilita la entrega del proyecto en un único entorno autocontenido lo que reduce costes de infraestructura, simplifica el despliegue y el mantenimiento. La herramienta sólo tiene conexión con GitHub.

El stack tecnológico, todo de código abierto, seleccionado es el siguiente. :

- **Python 3.14:** Lenguaje principal. Destaca su eficiencia en el procesamiento de cadenas de texto y su modelo de concurrencia asíncrona, que permite manejar múltiples tareas a la vez.
- **FastAPI:** Framework para la capa de servicios REST. Se apoya en Starlette y Uvicorn; además, incorpora validación de datos mediante tipos estáticos con Pydantic lo que simplifica las peticiones de auditoría.
- **GitPython.** Interfaz sobre Git. Le permite al sistema conectarse a repositorios remotos como los de GitHub clonarlos y recorrer la estructura archivos, ya sea cargándolos en memoria o en directorios temporales.
- **SQLAlchemy:** ORM que separa la lógica de negocio del SQL facilita el mantenimiento del código y refuerza la correspondencia entre las clases del dominio y las tablas de la base de datos.
- **SQLite 3:** Motor relacional integrado que concentra la información en un archivo único (*data/cmmi_auditor.db*). Evita los costes y la complejidad de mantener un servidor de bases de datos dedicado.
- **HTML5 / CSS3 / JavaScript (Vanilla):** Tecnologías nativas del navegador para la interfaz de usuario. Al prescindir de librerías o frameworks externos, la carga es más rápida y no hay dependencias de terceros

5.3. Requisitos del sistema

5.3.1. Requisitos No funcionales:

Los requisitos no funcionales definen los atributos de calidad, restricciones técnicas y niveles de eficiencia que determinan la viabilidad operativa y la seguridad de la solución. Para que el funcionamiento del sistema sea estable, se establecen restricciones de comportamiento, que se han adoptado siguiendo el modelo de calidad de McCall (Reynoso et al., 2010) .Los umbrales definidos se justifican mediante teorías consolidadas de ingeniería del software y se identifican en la tabla 27.

Perspectiva de Calidad (McCall) [30]	Factor de Calidad	Código	Especificación del Requisito No Funcional	Fundamentos
Perspectiva de operación (Eficacia en condiciones normales de funcionamiento)	Eficiencia / Rendimiento	RNF-01	El análisis estático de texto plano, la compilación de expresiones regulares (Regex) y el cálculo de cumplimiento de la PA de RDM deben completarse en un tiempo máximo de 2.5 segundos tras finalizar el clonado del repositorio de Git.	(Criterios de infraestructura sin fricción en II 2.4)(CMMI Institute, 2018; Nielsen, 1993; <i>Response Time Limits</i> , s.f.)
	Facilidad de Uso (Usabilidad)	RNF-02	El sistema debe exponer una interfaz web interactiva que exponga los hallazgos métricos de CMMI mediante anclas visuales normalizadas (Verde/Rojo) de fácil interpretación.	(UI, 2024)
	Integridad (Seguridad)	RNF-03	Las operaciones de clonado temporal de Git de repositorios de GitHub deben ejecutarse de forma aislada, aplicando políticas restrictivas de solo lectura (<i>read-only</i>) sobre los artefactos analizados.	(McGraw, 2006)
Perspectiva de Revisión (Capacidad de adaptación a cambios futuros)	Flexibilidad / Modularidad	RNF-04	La arquitectura del software debe permitir inyectar o remover dinámicamente nuevas Áreas de Práctica (PAs) de CMMI mediante la modificación exclusiva del manifiesto estructura.txt.	Hipótesis inicial de diseño. Modularidad del sistema

	Facilidad de Prueba (Testabilidad)	RNF-05	Cada módulo que aloje reglas semánticas de CMMI (como <code>rdm_scrum.py</code>) debe operar de forma aislada, permitiendo la ejecución independiente de pruebas unitarias sobre sus expresiones regulares.	(Ramzy, 2026)
Perspectiva de Transición (Capacidad de adaptación a diferentes entornos)	Portabilidad	RNF-06	El artefacto software debe ser completamente autónomo, liviano y capaz de ejecutarse sobre una base de datos relacional embebida (SQLite 3) sin requerir la preinstalación de servidores de bases de datos externos.	(<i>Def-Guide-to-Sqlite/2829.Pdf at Master · Apress/Def-Guide-to-Sqlite</i> , s. f.)
	Interoperabilidad	RNF-07	El <i>backend</i> de la aplicación debe exponer sus servicios funcionales mediante una API REST estándar bajo el <i>framework</i> FastAPI, estructurando las respuestas del motor en formato universal JSON.	(ISO 25010, s.f.; ISO IEC 25010 2011 <i>Systems and Software Quality Requirements and Evaluation SQuaRE Quality Model.pdf</i> , s.f.)

TABLA 27. REQUISITOS NO FUNCIONALES SEGÚN LAS PERSPECTIVAS DE MCCALL

5.3.2. Requisitos funcionales

Los requisitos funcionales describen qué debe hacer el sistema. La herramienta diseñada es un motor de auditoría CMMI. Su función es verificar de forma automatizada que los artefactos de Scrum cumplen con las prácticas del modelo.

Las prácticas cubiertas en la PoC pertenecen al área de Desarrollo y Gestión de Requisitos (RDM), salvo el cálculo de métricas, que se apoya en Medición y Gestión del Desempeño (MPM). Cada uno se asocia a una o más prácticas del Nivel de Madurez 2 de CMMI-DEV V2.0. La Tabla 28 recoge esta relación.

RF	Descripción	Práctica CMMI	Archivo de código
RF-01	Clonación y lectura del repositorio local	RDM 1.1 / CM 2.2	GitPython_Repo / motor_auditoria.py
RF-02	Indexación de historias de usuario	RDM 1.1	rdm_scrum.py
RF-03	Validación de la narrativa ágil	RDM 2.1 / RDM 2.2	rdm_scrum.py
RF-04	Validación de criterios de aceptación	RDM 2.2	rdm_scrum.py
RF-05	Auditoría de la <i>Definition of Done</i>	RDM 2.6	rdm_scrum.py
RF-06	Cálculo del porcentaje de cumplimiento	MPM 2.3 / MPM 2.5	motor_auditoria.py

TABLA 28. TRAZABILIDAD ENTRE RF, PRÁCTICAS CMMI Y ARCHIVOS DE CÓDIGO.

A continuación, se describe cada RF, junto con el fragmento de código que lo implementa.

RF-01. Clonación y lectura del repositorio local: La librería GitPython queda encapsulada en la clase GitPython_Repo (subsistema Git). Esta se conecta por HTTPS a la API de GitHub, elimina el directorio temporal mediante `shutil.rmtree` y clona el repositorio en `data/test_repo/`. El motor de auditoría lee dicho repositorio local y audita el proyecto sin modificar el código original. Este RF se asocia con las prácticas:

- RDM 1.1 (Registrar requisitos): El motor cumple la regla porque descarga y centraliza todo el *Product Backlog* (los requisitos) en un solo lugar seguro y accesible para poder revisarlo.
- CM 2.2 (Control de configuración): El motor cumple la regla porque nunca trabaja sobre el código real del cliente. Crea una copia temporal aislada y la elimina al terminar.

Y se implementa en el script: (`src/auditor_calculation/motor_auditoria.py`):

```
def ejecutar_auditoria(self, repo_path: str, modulo: ModuloRDMScrum,
verbose: bool = False) -> Dict[str, Any]:
    """
    Ejecuta una auditoría completa sobre un repositorio.

    Args:
        repo_path: Ruta al repositorio a auditar
        modulo: Módulo de evaluación (ej: ModuloRDMScrum)
        verbose: Mostrar detalles en consola

    Returns:
        Diccionario con resultados completos de la auditoría
    """
    print(f"🔍 Iniciando auditoría en: {repo_path}")

    # 1. Evaluar con el módulo
    resultado = modulo.evaluar(repo_path, verbose=verbose)
```

RF-02. Indexación de las historias de usuario: El motor recorre los archivos Markdown del repositorio y localiza las historias de usuario. Para ello busca archivos extensión `.md` y, dentro de ellos, encabezados que sigan el patrón `US-`. Se conecta con la práctica RDM 1.1. El sistema cumple esta práctica porque centraliza y archiva todas las historias de usuario en la base de datos. Esto garantiza que la empresa disponga de un inventario único, formal y actualizado con todas las necesidades del cliente. Y se implementa en (`src/cmml_rules/rdm_scrum.py`)

```

"RDM_1_1": {
  "nombre": "Registro de requisitos",
  "nivel": 1,
  "peso": 15,
  "archivos": [
    "product_backlog.md", "backlog.csv", "requirements.md",
    "README.md", "REQUIREMENTS.md", "user_stories.md"
  ],
  "patrones_alternativos": [
    r"(?i)requisitos?:", r"(?i)requirements:",
    r"(?i)user\s+stories?:", r"(?i)backlog:"
  ],
  "descripcion": "Los requisitos deben estar registrados en archivos accesibles"
},

```

RF-03. Validación de la narrativa ágil: El motor comprueba que cada historia de usuario siga la estructura "Como... quiero... para...". Acepta dos variantes: con comas entre los bloques y sin ellas, como si decide desglosarlas en líneas independientes. Asegura el cumplimiento de las prácticas RDM 2.1 (Obtener necesidades de los *stakeholders*) y RDM 2.2 (Transformarlas en requisitos priorizados) al asegurar que los requerimientos del cliente no sean simples ideas sueltas. Al exigir una estructura formal, el sistema garantiza que se identifique formalmente al actor (*stakeholder*), la necesidad técnica real y el valor de negocio que justifica el desarrollo. Se implementa en (src/cmimi_rules/rdm_scrum.py):

```

"RDM_2_1": {
  "nombre": "Entendimiento de necesidades de stakeholders",
  "nivel": 2,
  "peso": 15,
  "patrones_alternativos": [
    # Formato con comas: "Como X, Quiero Y, Para Z"
    r"(?i)como\s+[\w\s\(\)\-]+\s*quiero\s+[\w\s\(\)\-]+\s*para\s+[\w\s\(\)\-]+",
    # Formato sin comas: "Como X\nQuiero Y\nPara Z"
    r"(?i)como\s+[\w\s\(\)\-]+\s*\r\n\s*quiero\s+[\w\s\(\)\-]+\s*\r\n\s*para\s+[\w\s\(\)\-]+",
    # Palabras clave sueltas
    r"(?i)stakeholder\s+need", r"(?i)necesidad\s+de",
    r"(?i)historias?\s+de\s+usuario", r"(?i)user\s+stories?"
  ],
  "descripcion": "Las necesidades de stakeholders deben estar documentadas (formato 'Como... Quiero... Para...')",
},
"RDM_2_2": {
  "nombre": "Transformación en requisitos priorizados",
  "nivel": 2,
  "peso": 15,
  "patrones_alternativos": [
    r"(?i)criterios?\s+de\s+aceptaci[oó]n",
    r"(?i)acceptance\s+criteria",
    r"(?i)prioridad", r"(?i)priority",
    r"(?i)must\s+have", r"(?i)should\s+have", r"(?i)could\s+have",
    r"(?i)prioridad\s*:",
    r"(?i)priority\s*:"
  ],
  "descripcion": "Los requisitos deben tener criterios de aceptación y priorización"
},

```

RF-04. Validación de los criterios de aceptación: El motor revisa que cada historia de usuario incluya criterios de aceptación y una prioridad. Si no encuentra ninguno de los dos, la auditoría del proyecto falla. Se asocia con la práctica RDM 2.2 (Establecer criterios de aceptación de los requisitos). El software cumple esta regla porque evita que el equipo trabaje con requisitos ambiguos o incompletos. Al obligar a escribir las condiciones de aceptación y la prioridad, el sistema asegura que los desarrolladores y el cliente tengan un acuerdo claro y objetivo sobre qué requisitos son urgentes y qué condiciones exactas debe cumplir el software para darlo por aprobado. Se implementa en (src/cmmi_rules/rdm_scrum.py):

```
"RDM_2_2": {
  "nombre": "Transformación en requisitos priorizados",
  "nivel": 2,
  "peso": 15,
  "patrones_alternativos": [
    r"(?i)criterios?\s+de\s+aceptaci[oó]n",
    r"(?i)acceptance\s+criteria",
    r"(?i)prioridad", r"(?i)priority",
    r"(?i)must\s+have", r"(?i)should\s+have", r"(?i)could\s+have",
    r"(?i)prioridad\s*:",
    r"(?i)priority\s*:"
  ],
  "descripcion": "Los requisitos deben tener criterios de aceptación y priorización"
},
```

RF-05. Auditoría de la Definition of Done: El programa verifica que exista el archivo maestro con los estándares en el repositorio. El motor ejecuta una búsqueda para localizar ficheros con nombres clave: `DOD.md`, `DEFINITION_OF_DONE.md` o `definition_of_done.md` y analiza que el documento contenga evidencias de cuatro controles de calidad obligatorios:

Protocolos de Verificación: El motor rastrea términos como *verificación*, *verification* o *checklist* para verificar el uso de métodos sistemáticos de revisión de software antes de cada entrega.

Criterios de Idoneidad: Busca cadenas de texto como *definition of done*, *dod* o *done* para validar que el documento sea una guía de fin de tarea.

Consistencia de Procesos: Busca las palabras como *consistencia*, *consistency* o *coherencia* para certificar que las tareas se alinean con las metas del proyecto.

Matrices de Seguimiento: Busca patrones como *traceability matrix* o *trazabilidad*, para verificar que existe una conexión entre los casos de prueba y las historias de usuario.

Se asocia a RDM 2.6 (Asegurar la consistencia entre los requisitos y los productos de trabajo) y se implementa en (`src/cmmi_rules/rdm_scrum.py`):

```
"RDM_2_6": {
    "nombre": "Consistencia entre requisitos, planes y productos de trabajo",
    "nivel": 2,
    "peso": 15,
    "archivos": [
        "DOD.md", "DEFINITION_OF_DONE.md", "definition_of_done.md",
        "CONSISTENCY_CHECK.md", "VERIFICATION_REPORT.md"
    ],
    "patrones_alternativos": [
        r"(?i)definition\s+of\s+done", r"(?i)dod",
        r"(?i)checklist", r"(?i)verificad[oó]n",
        r"(?i)verification", r"(?i)consistency",
        r"(?i)consistencia", r"(?i)traceability\s+matrix"
    ],
    "descripcion": "Debe existir verificación de consistencia entre requisitos, planes y productos de trabajo"
}
```

RF-06. Cálculo del porcentaje de cumplimiento: El motor reúne todos los datos de la revisión mediante una función interna (`calcular_metricas`). Cuenta los criterios y reglas analizados, clasifica las alertas según dos niveles de gravedad (nivel 1 para fallos críticos y nivel 2 para simples mejoras de código) y envía y almacena una estadística. Y guarda la fecha y hora de auditoría. Se asocia a las prácticas MPM 2.3 (Obtener y registrar datos de métricas) y MPM 2.5 (Almacenar datos de medición). Si no hay historias, no hay criterios o no hay DoD, la auditoria es negativa. Se implementa en (`src/auditor_calculation/motor_auditoria.py`):

```
def _calcular_metricas(self, resultado: Dict) -> Dict:
    """Calcula métricas adicionales del resultado."""
    metricas = {
        "tiempo_ejecucion": datetime.utcnow().isoformat(),
        "criterios_evaluados": len(resultado.get("criterios", {})),
        "porcentaje_cumplimiento": resultado.get("porcentaje_cumplimiento", 0),
        "hallazgos_criticos": len([h for h in resultado.get("hallazgos", []) if h.get("nivel") == 1]),
        "hallazgos_mejora": len([h for h in resultado.get("hallazgos", []) if h.get("nivel") == 2]),
        "recomendaciones": len(resultado.get("recomendaciones_generales", []))
    }
```

5.4. Descripción funcional:

5.4.1. Arquitectura funcional:

Se describe la arquitectura del sistema siguiendo el modelo propuesto por Krutchen en su modelo de 4+1 vistas y que atienden a las diferentes necesidades de los distintos *stakeholders* (Krutchen, s.f.).

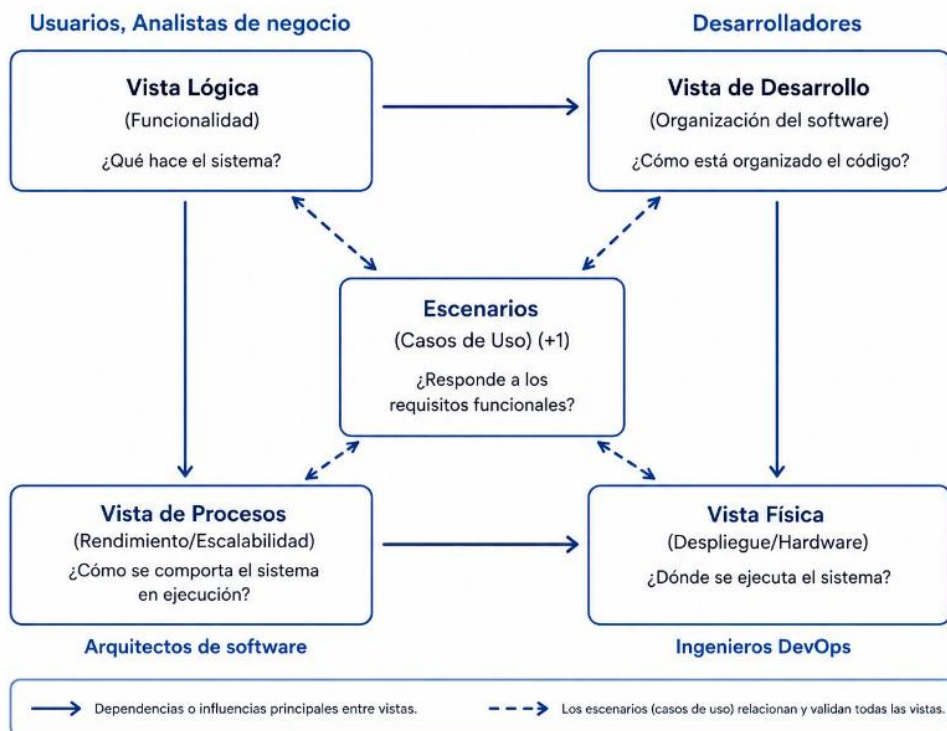


ILUSTRACIÓN 9. MODELO DE KRUTCHEN. ELABORACIÓN PROPIA

Atendiendo al modelo se representan las vistas mediante diagramas UML, (Lenguaje de Modelado Unificado). Se han introducido sólo los diagramas principales, para evitar redundancias.

Vista física. Diagrama de Despliegue:

La Ilustración 10, muestra la distribución del hardware y la infraestructura donde se ejecuta la solución. La aplicación se ejecuta en una estación de trabajo local sobre un entorno virtual de Python 3.14. El servidor web Uvicorn levanta la aplicación en el puerto local 127.0.0.1:8000. La aplicación lee el archivo de configuración `estructura.txt` y escribe en la base de datos SQLite a través del sistema de archivos local. El subsistema

Git se conecta por HTTPS a la API de GitHub para clonar los repositorios. El usuario auditor accede al sistema desde un navegador web mediante peticiones HTTP/JSON.

Vista de Desarrollo: Diagrama de Componentes

La Ilustración 11, muestra la organización de los módulos y sus dependencias. El Manifiesto de Configuración (*estructura.txt*) alimenta los metadatos de las áreas a la Capa de Ingesta (*seed_data.py*), que puebla las tablas base de la base de datos. El Dashboard se comunica con la API REST de FastAPI. Cuando se lanza una petición de auditoría, el Motor de Auditoría recupera los datos del repositorio y carga las reglas del (*rdm_scrum.py*) y le ordena al Subsistema Git que descargue y analice los archivos .md del repositorio remoto GitHub.

Vista de Lógica: Diagrama de Clases

La Ilustración 12 muestra la estructura orientada a objetos. El núcleo lo forman la configuración de la aplicación y la sesión de SQLAlchemy (*DatabaseConfig*). El enrutador de FastAPI expone los *endpoints*. La clase principal de negocio, MotorAuditoria, se conecta directamente con *Patrones_Busqueda_RDM* (clase encargada de compilar las expresiones regulares) y con la interfaz de GitPython (*GitPython_Repo*). Las clases del modelo de datos ORM: *Proyecto*, *ResultadoAuditoria*, *CriterioRDM* y *CumplimientoCriterio*, se vinculan mediante relaciones de integridad referencial (claves foráneas) de uno a muchos (1: N), mapeando de forma exacta la trazabilidad exigida por la práctica CMMI RDM 2.6.

Vista de Procesos: Diagrama de Actividad

La Ilustración 13 muestra el flujo de ejecución. Al acceder al *dashboard*, el sistema consulta si hay proyectos registrados en SQLite y los muestra en un listado. Cuando el usuario selecciona un repositorio y pulsa 'Ejecutar Auditoría', el *backend* activa el motor: clona el repositorio de GitHub, analiza las expresiones regulares sobre el backlog, los compromisos y la DoD, calcula los porcentajes de cumplimiento y guarda los hallazgos en la base de datos. La API devuelve un objeto JSON al *frontend*, que actualiza los gráficos y las métricas en pantalla.

Vista de Escenarios: Diagrama de Casos de Uso (+1)

La Ilustración 14 muestra cómo interactúan los actores con el sistema. El *Product Owner* gestiona el registro y los cambios del backlog (RDM 1.1 y 2.5). Los *Developers* definen las historias y validan los incrementos frente a la DoD (RDM 2.1, 2.2 y 2.6). El Scrum Master supervisa las auditorías automáticas de configuración. Los actores externos (Auditor Interno y *Lead Appraiser*) consultan las métricas, el historial de trazabilidad y los informes consolidados.

Diagrama de Despliegue: Entorno Local de Auditoría y Conexión Remota a GitHub

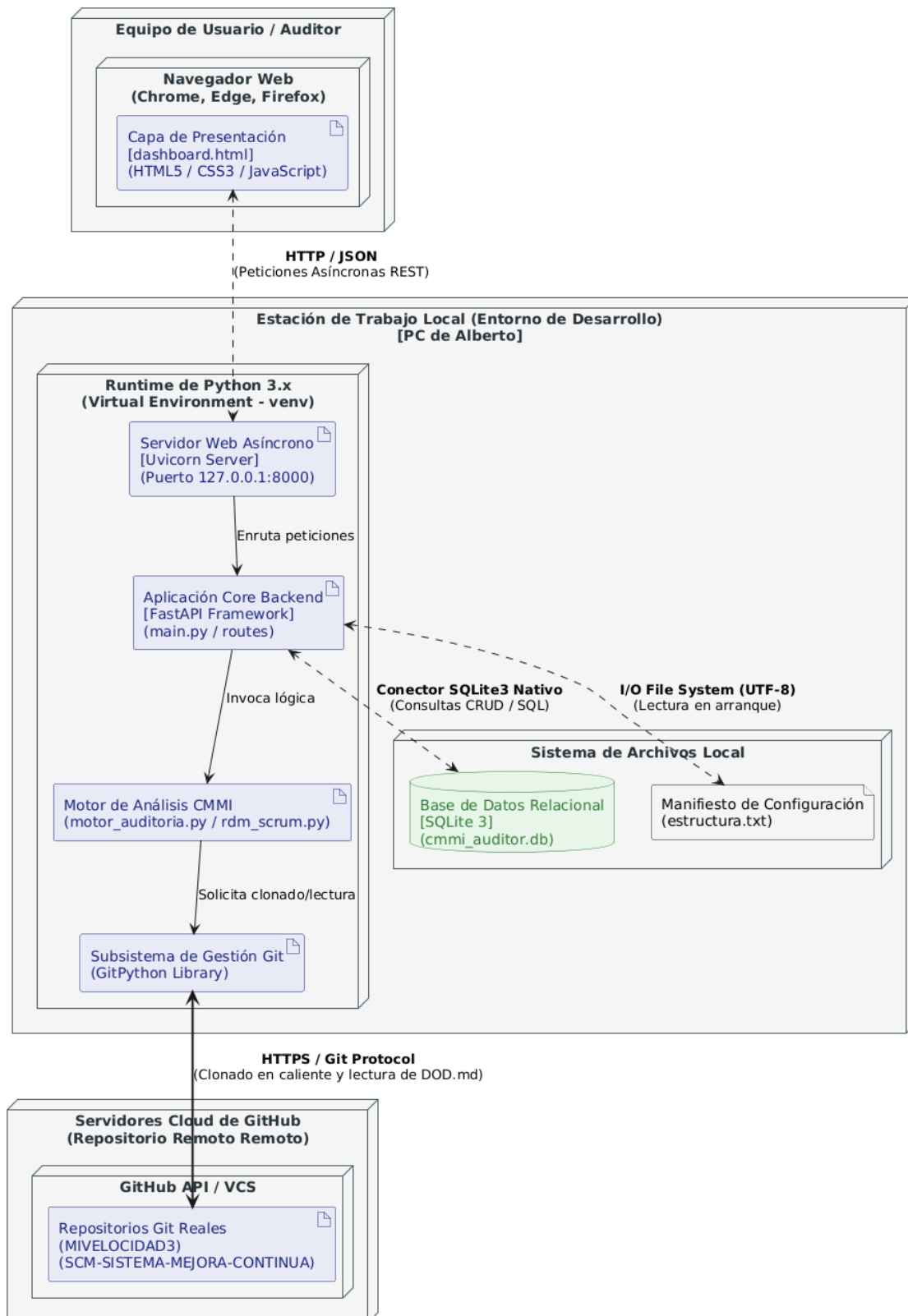


ILUSTRACIÓN 10. VISTA FÍSICA. DIAGRAMA DE DESPLIEGUE. ELABORACIÓN PROPIA

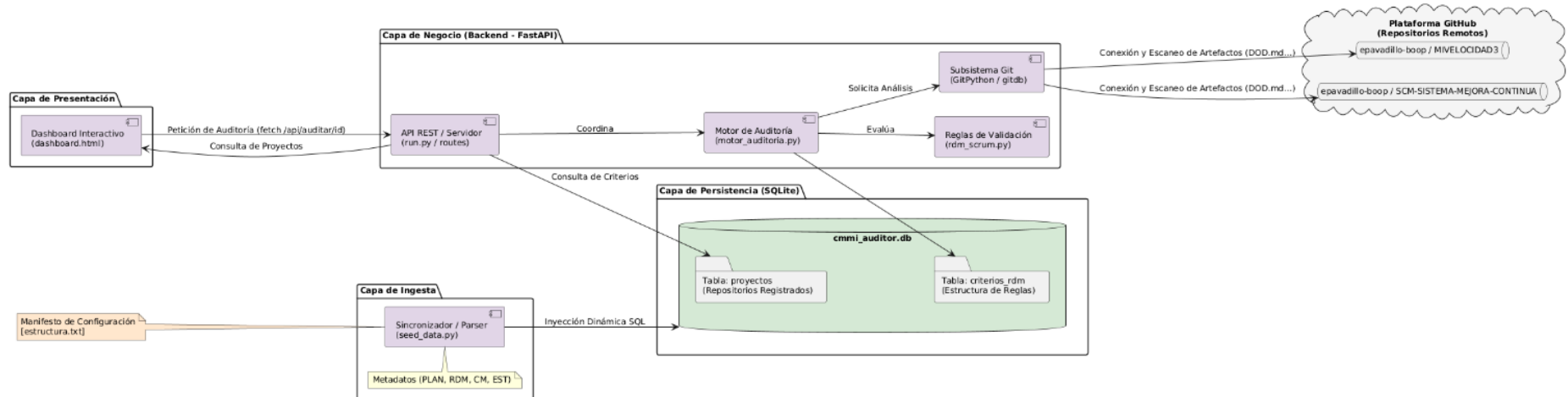


ILUSTRACIÓN 11. VISTA DE DESARROLLO. DIAGRAMA DE COMPONENTES. ELABORACIÓN PROPIA

Diagrama de Clases - Arquitectura Modular PoC

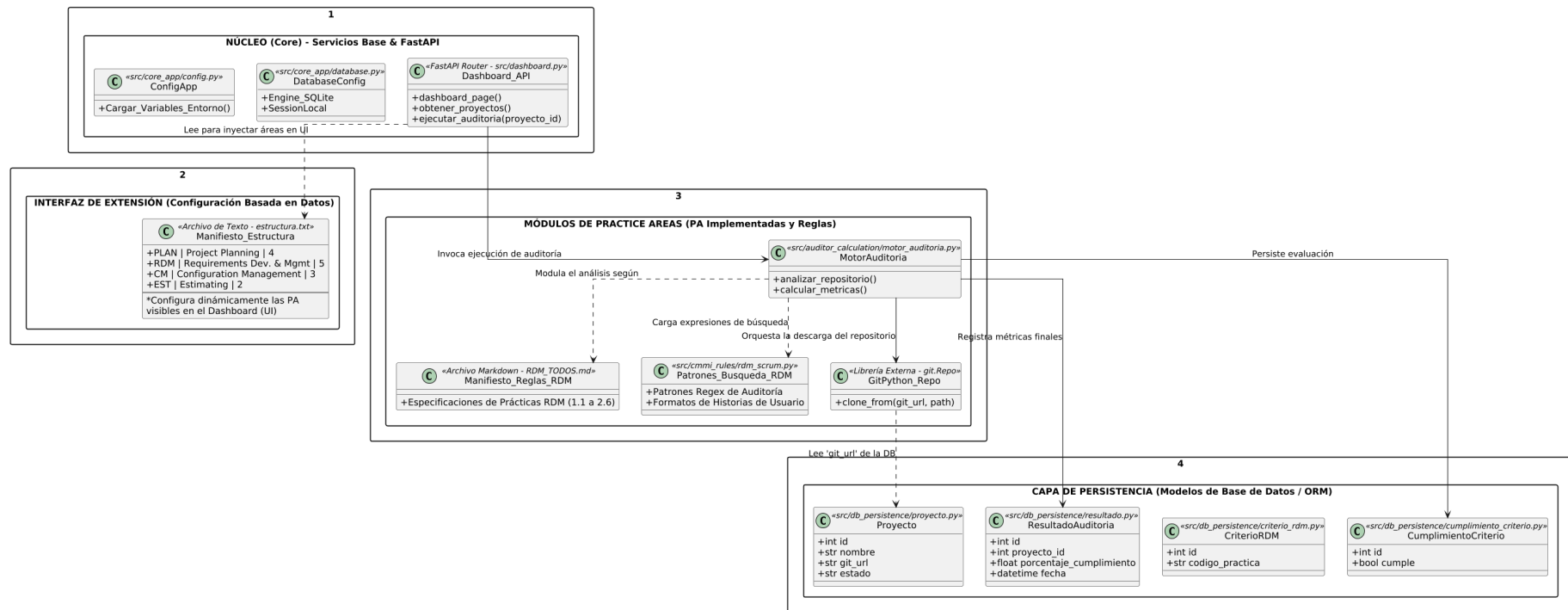


ILUSTRACIÓN 12. VISTA LÓGICA. DIAGRAMA DE CLASES. ELABORACIÓN PROPIA

Diagrama de Actividad: Ciclo Completo del Proceso de Auditoría Git (TFM)

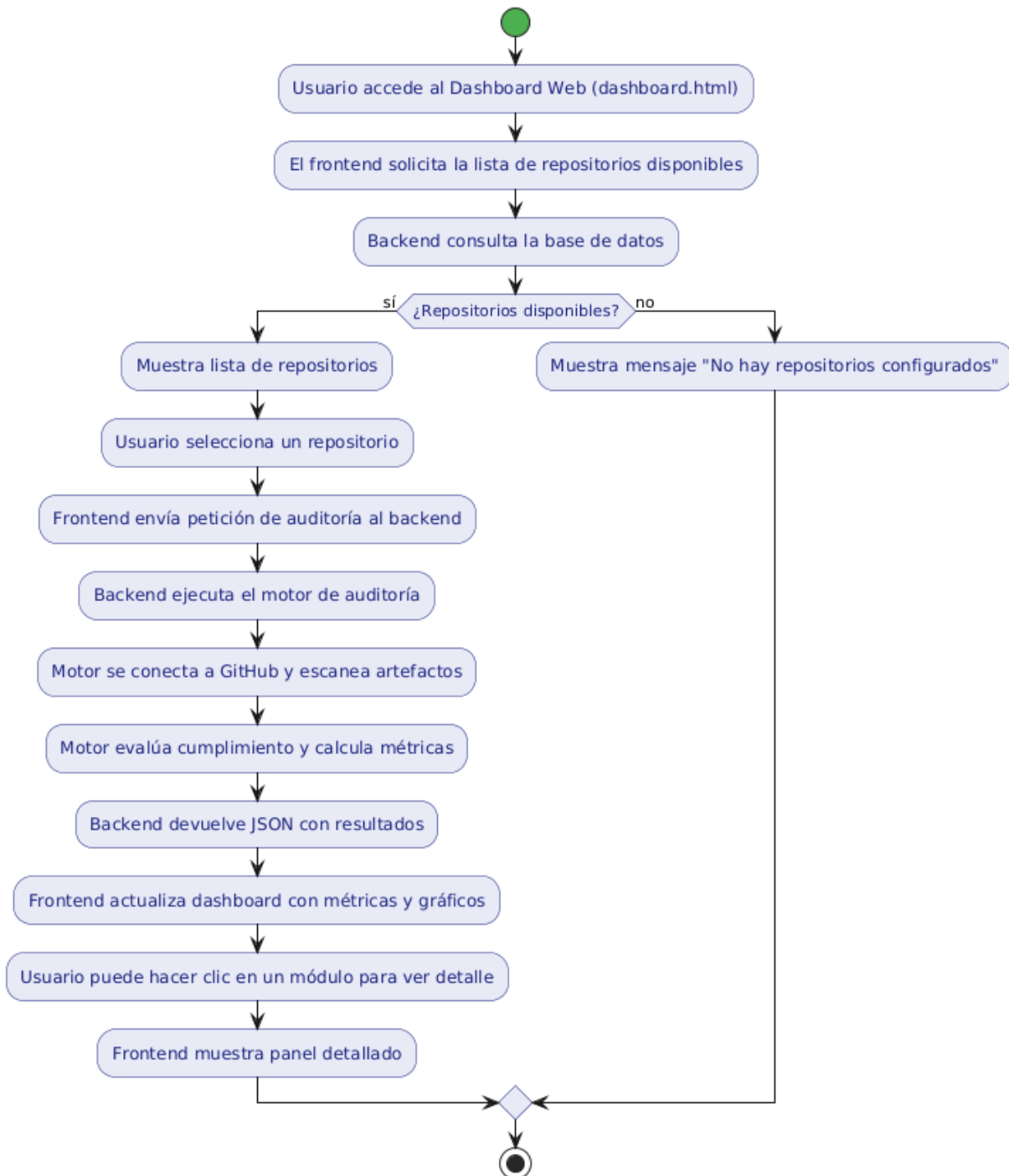


ILUSTRACIÓN 13. VISTA DE PROCESOS. DIAGRAMA DE ACTIVIDAD. ELABORACIÓN PROPIA

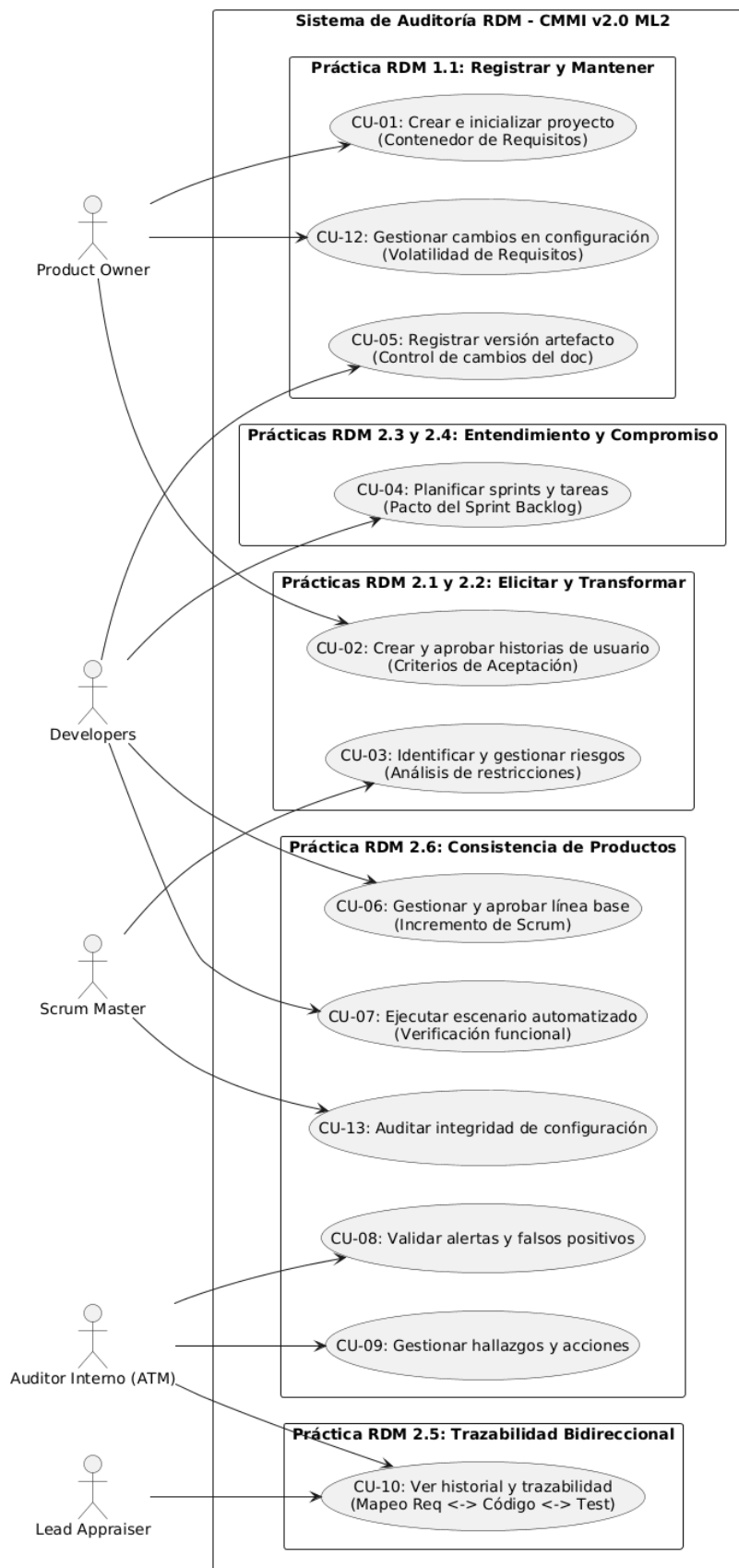


ILUSTRACIÓN 14.VISTA DE CASOS DE USO. ELABORACIÓN PROPIA

Matriz de Casos de Uso (CU-01 a CU-13). Enfoque CMMI v2.0 ML2 (RDM)

Código	Nombre	Actores	Flujo resumido	Evidencias	Patrón de búsqueda
CU-01	Crear e inicializar proyecto	PO / SM	1. Define nombre y objetivo. 2. Crea carpetas y archivos. 3. Configura el repositorio. 4. Define roles y políticas.	README.md, estructura de directorios, product_backlog.md inicial	No se definen en el alcance de la PoC
CU-02	Crear y aprobar historias de usuario	PO / Dev	1. Identifica la necesidad. 2. Redacta la historia (Como/Quiero/Para). 3. Define criterios de aceptación. 4. Prioriza y aprueba.	Historia en backlog, criterios de aceptación, prioridad, aprobación del PO	Narrativa: como\s+[\w\s]+,\s*quiero\s+[\w\s]+,\s*para\s+[\w\s]+. Criterios: criterios?\s+de\s+aceptaci[oó]n, acceptance\s+criteria. Priorización: prioridad, priority, must\s+have, should\s+have, could\s+have
CU-03	Identificar y gestionar riesgos	SM / PO	1. Identifica amenazas. 2. Analiza probabilidad e impacto. 3. Define planes de	Lista de riesgos, análisis de impacto, planes de mitigación	No se definen en el alcance de la PoC

Código	Nombre	Actores	Flujo resumido	Evidencias	Patrón de búsqueda
			mitigación. 4. Monitoriza.		
CU-04	Planificar <i>sprints</i> y tareas	Dev / SM	1. Selecciona historias. 2. Descompone en tareas. 3. Estima esfuerzo. 4. Asume el compromiso.	Sprint backlog, estimaciones, capacidad, compromiso	Archivos: SPRINT_GOAL.md, COMMITMENTS.md, PLANNING_POKER.md, SPRINT_PLANNING.md, sprint_planning.md, SPRINT_COMMITMENTS.md. Patrones: sprint\s+goal, planning\s+poker, commitment, compromiso, estimaci[oón], estimation, story\s+points?
CU-05	Registrar versión de artefacto	Dev / SM	1. Selecciona el artefacto. 2. Asigna identificador de versión. 3. Guarda cambios en Git. 4. Registra el estado.	Artefactos en control, ID únicos, registro de versiones	No se definen en el alcance de la PoC
CU-06	Gestionar y aprobar línea base	Dev / PO	1. Identifica piezas de código. 2. Revisa integridad con CI/CD. 3. Crea tag o	Línea base documentada, aprobación, lista de artefactos	No se definen en el alcance de la PoC

Código	Nombre	Actores	Flujo resumido	Evidencias	Patrón de búsqueda
			<i>release</i> . 4. Obtiene aval del PO.		
CU-07	Ejecutar escenario automatizado	Dev / SM	1. El motor selecciona escenarios. 2. Ejecuta pruebas. 3. Evalúa fallos y genera alertas.	Resultados automatizados, reportes, alertas, trazabilidad requisito-prueba	Patrones: (US REQ STORY TASK)-\d{3,4}, commit\s+[a-f0-9]{7,40}, test, prueba, traceability, trazabilidad, matriz\s+de\s+trazabilidad
CU-08	Validar alertas y falsos positivos	ATM / Dev	1. Revisa la alerta. 2. Verifica si hay desvío real. 3. Descarta o confirma el fallo.	Registro de alertas, documentos de falsos positivos, hallazgos	No se definen en el alcance de la PoC
CU-09	Gestionar hallazgos y acciones	ATM / SM	1. Registra el desvío. 2. Asigna responsable. 3. Traza plan de acción. 4. Valida y cierra.	Lista de hallazgos, planes de acción, evidencia de cierre	No se definen en el alcance de la PoC

Código	Nombre	Actores	Flujo resumido	Evidencias	Patrón de búsqueda
CU-10	Ver historial y trazabilidad	ATM / Lead Appraiser	1. Selecciona la <i>User Story</i> . 2. Rastrea su evolución en <i>commits</i> . 3. Vincula requisito con prueba.	Historial de cambios, matriz de trazabilidad, registro de versiones	Patrones: (US REQ STORY TASK)-\d{3,4}, commit\s+[a-f0-9]{7,40}, test, prueba, traceability, trazabilidad, matriz\s+de\s+trazabilidad
CU-11	Firmar y emitir informe	Lead Appraiser / Alta Dirección	1. Consolida resultados. 2. Revisa con el equipo. 3. Firma y distribuye el dictamen.	Informe final firmado, registro de emisión	No se definen en el alcance de la PoC
CU-12	Gestionar cambios en configuración	Dev / PO	1. Recibe solicitud de cambio. 2. Evalúa impacto técnico. 3. El PO aprueba o rechaza. 4. Modifica la <i>User Story</i> .	RFC, análisis de impacto, <i>commits</i> , <i>backlog</i> ajustado	No se definen en el alcance de la PoC

Código	Nombre	Actores	Flujo resumido	Evidencias	Patrón de búsqueda
CU-13	Auditar integridad de configuración	SM / ATM	1. Revisa ítems del <i>release</i> . 2. Compara con la DoD. 3. Valida consistencia backlog-producción.	Reporte de auditoría CM, <i>checklist</i> DoD, registro de desvíos	Archivos: DOD.md, DEFINITION_OF_DONE.md, definition_of_done.md, CONSISTENCY_CHECK.md, VERIFICATION_REPORT.md. Patrones: definition\s+of\s+done, dod, checklist, verificad[oó]n, verification, consistency, consistencia, traceability\s+matrix

TABLA 29. MATRIZ CASOS USO. ENFOQUE CMMI DEV 2.0 ML2 RDM

5.5. Desarrollo de la Prueba de Concepto (PoC)

5.5.1. Arquitectura de despliegue.

La aplicación separa la lógica de negocio de la infraestructura de red. Esto permite modificar la configuración del entorno sin alterar las reglas del auditor. El despliegue se organiza en cuatro componentes:

- `run.py`. En la raíz del proyecto, inicializa el servidor Uvicorn en local. Configura host, puerto y modo de recarga, y apunta a la instancia de la aplicación empaquetada en `src/`.
- `src/dashboard.py` y `src/login.py`. Aquí se define la API: la lógica de la interfaz web, el enrutamiento de vistas, las dependencias y la seguridad.
- `src/auditor_calculation/` y `src/cmmi_rules/`. El motor de cálculo y las reglas CMMI operan con independencia de la capa web. Se conectan a la base de datos SQLite configurada en `src/core_app/database.py`.
- `app/webui/`. Contiene los estilos y la maquetación de la interfaz gráfica. Se mantiene desacoplado de la lógica de la API, de modo que los cambios en la presentación no afectan al enrutamiento ni a las reglas del auditor.

La estructura de archivos es la siguiente:

```
cmmi-auditor.poc/
├── app/
│   └── webui/                # Interfaz de usuario (HTML/CSS)
├── data/
│   └── cmmi_auditor.db       # Base de datos SQLite del sistema
├── src/
│   ├── auditor_calculation/  # Núcleo del motor de auditoría
│   ├── cmmi_rules/           # Reglas CMMI implementadas
│   ├── core_app/             # Configuraciones núcleo y base de datos
│   ├── db_persistence/       # Modelos de persistencia
│   ├── dashboard.py          # Dashboard de la aplicación FastAPI
│   └── login.py              # Gestión de sesiones y login FastAPI
├── .env                      # Variables de entorno y configuración
├── requirements.txt          # Dependencias del proyecto
└── run.py                    # Script de inicialización de Uvicorn
```

ILUSTRACIÓN 15. ESTRUCTURA DE ARCHIVOS DESPLIEGUE CMMI-AUDITOR POC

Flujo de Ejecución

El flujo de ejecución es el siguiente:

1. • `run.py`. En la raíz del proyecto, inicializa el servidor Uvicorn en local. Configura host, puerto y modo de recarga, y apunta a la instancia de la aplicación empaquetada en `src/`.
2. • `src/dashboard.py` y `src/login.py`. Aquí se define la API: la lógica de la interfaz web, el enrutamiento de vistas, las dependencias y la seguridad.
3. • `src/auditor_calculation/` y `src/cmmi_rules/`. El motor de cálculo y las reglas CMMI operan con independencia de la capa web. Se conectan a la base de datos SQLite configurada en `src/core_app/database.py`.
4. • `app/webui/`. Contiene los estilos y la maquetación de la interfaz gráfica. Se mantiene desacoplado de la lógica de la API, de esta forma los cambios en la presentación no afectan al enrutamiento ni a las reglas del auditor.

5.5.2. El manifiesto de configuración

El archivo `estructura.txt` define qué módulos de CMMI audita el sistema. Cada línea tiene tres campos separados por el carácter |: `Cod.PA|Nombre|NPrácticas`

- Código PA: Identificador único que se utiliza para enlazar con el módulo Python de reglas (por ejemplo, RDM se corresponde con `rdm_scrum.py`).
- Nombre: Etiqueta que se muestra en la interfaz.
- Número de prácticas Nivel 2: Cantidad de prácticas que deben verificarse para considerar satisfecha una PA en el Nivel de Madurez 2. Este valor se emplea para calcular el porcentaje de cumplimiento (ej. si de 5 prácticas se validan 4, el progreso es del 80%).

El manifiesto utilizado en la PoC contiene:

```
PLAN|Project Planning|4
RDM|Requirements Dev. & Mgmt|5
CM|Configuration Management|3
EST|Estimating|2
```

Al añadir o eliminar una línea en este archivo, el sistema activa o desactiva el correspondiente módulo. Esta propiedad permite escalar la herramienta a nuevas áreas de CMMI de forma inmediata.

5.5.3. Refactorización Arquitectónica para Modularidad

Para demostrar modularidad fue necesario refactorizar las tres capas del software (persistencia, negocio y presentación):

1. **Capa de Configuración** Se creó `estructura.txt`, un archivo de metadatos independiente que actúa como el manifiesto del sistema. Su formato simple `PA|Nombre|NPrácticas`. Permite cambiar las áreas de práctica sin tocar el código fuente.
2. **Capa de Ingesta y Persistencia** `seed_data.py`: se reprogramó como parser dinámico: lee el manifiesto línea a línea, crea los módulos y mapea sus cabeceras en las tablas `proyectos` y `criterios_rdm` El número de prácticas inicializa el contador de criterios esperados.
3. **Capa de Negocio** El motor (`src/auditor_calculation/`) dejó de depender de flujos cableados. Ahora recibe solicitudes basadas en el identificador del proyecto y, dado un código de PA, carga el módulo de reglas correspondiente (`rdm_scrum.py`, `cm_rules.py`) y ejecuta las validaciones.
4. **Capa de Presentación** La interfaz (`app/webui/`) se rediseñó para eliminar elementos fijos: descubre los módulos disponibles consultando la base de datos y repinta los componentes de detalle, según los resultados obtenidos en la auditoría.

Archivo Modificado	Estado Inicial (Monolítico / Rígido)	Estado Refactorizado (Modular / Dinámico)	Justificación Arquitectónica
estructura.txt	No existía	Fichero de configuración y metadatos (PLAN ..., CM ...)	Permite cambiar las reglas del negocio sin tocar código fuente.
seed_data.py	Insertaba datos fijos (<i>hardcoded</i>) solo para el módulo RDM.	Script de sincronización automatizado. Lee el <code>estructura.txt</code> línea por línea y pobla la BD.	Automatiza la ingesta de datos. Traduce la configuración del usuario en registros relacionales dinámicos.
run.py/ Backend	Solo levantaba rutas y lógica cableada para un único módulo.	Capacidad de procesar parámetros dinámicos (/api/auditar/{id}).	Independencia del Core. El <i>backend</i> ya no sabe qué es "RDM" solo procesa reglas genéricas de Git.
dashboard.html	Pantalla estática fija que solo pintaba la tarjeta de RDM.	Interfaz guiada por datos con grilla multimodular.	Permite la visualización e interactividad en tiempo de ejecución de cualquier módulo inyectado.

TABLA 30. ARCHIVOS REFACTORIZADOS PARA MODULARIDAD

5.5.4. Demostración Práctica de la Modularidad

La validación práctica de modularidad se realizó mediante la inclusión nuevas PA en el manifiesto `estructura.txt`. A continuación, se muestran las capturas de pantalla que del funcionamiento del sistema en diferentes configuraciones:

Escenario 1: Dashboard con Módulo RDM (Configuración Inicial)

Las Ilustración 16 y la Ilustración 17 muestran el *dashboard* configurado exclusivamente con el módulo RDM (Desarrollo y Gestión de Requisitos), que fue el punto de partida de la implementación, en la imagen puede apreciarse los diferentes resultados para cada proyecto seleccionado.

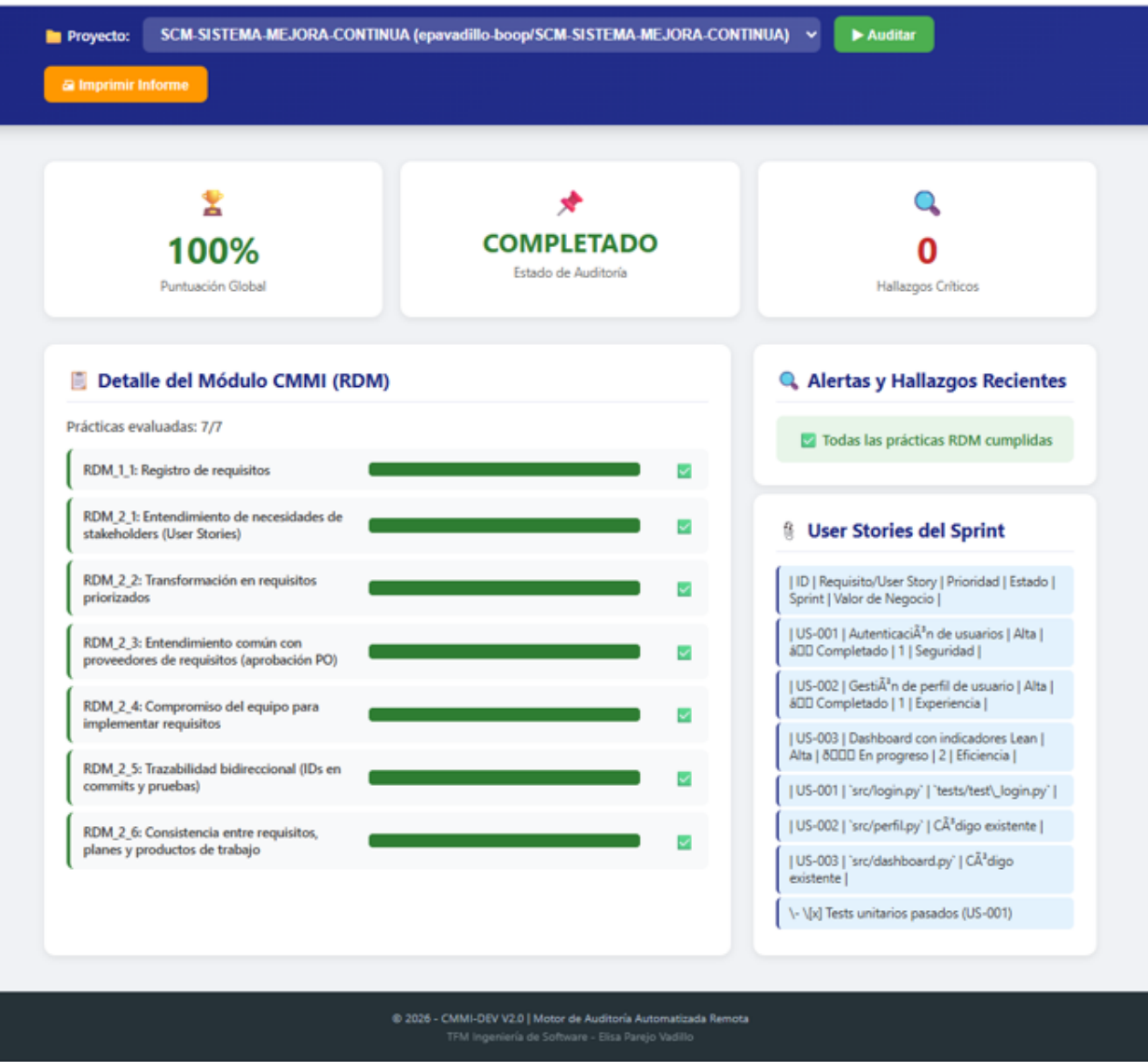


ILUSTRACIÓN 16. DASHBOARD CON CONFIGURACIÓN INICIAL PROYECTO SCM (RDM)

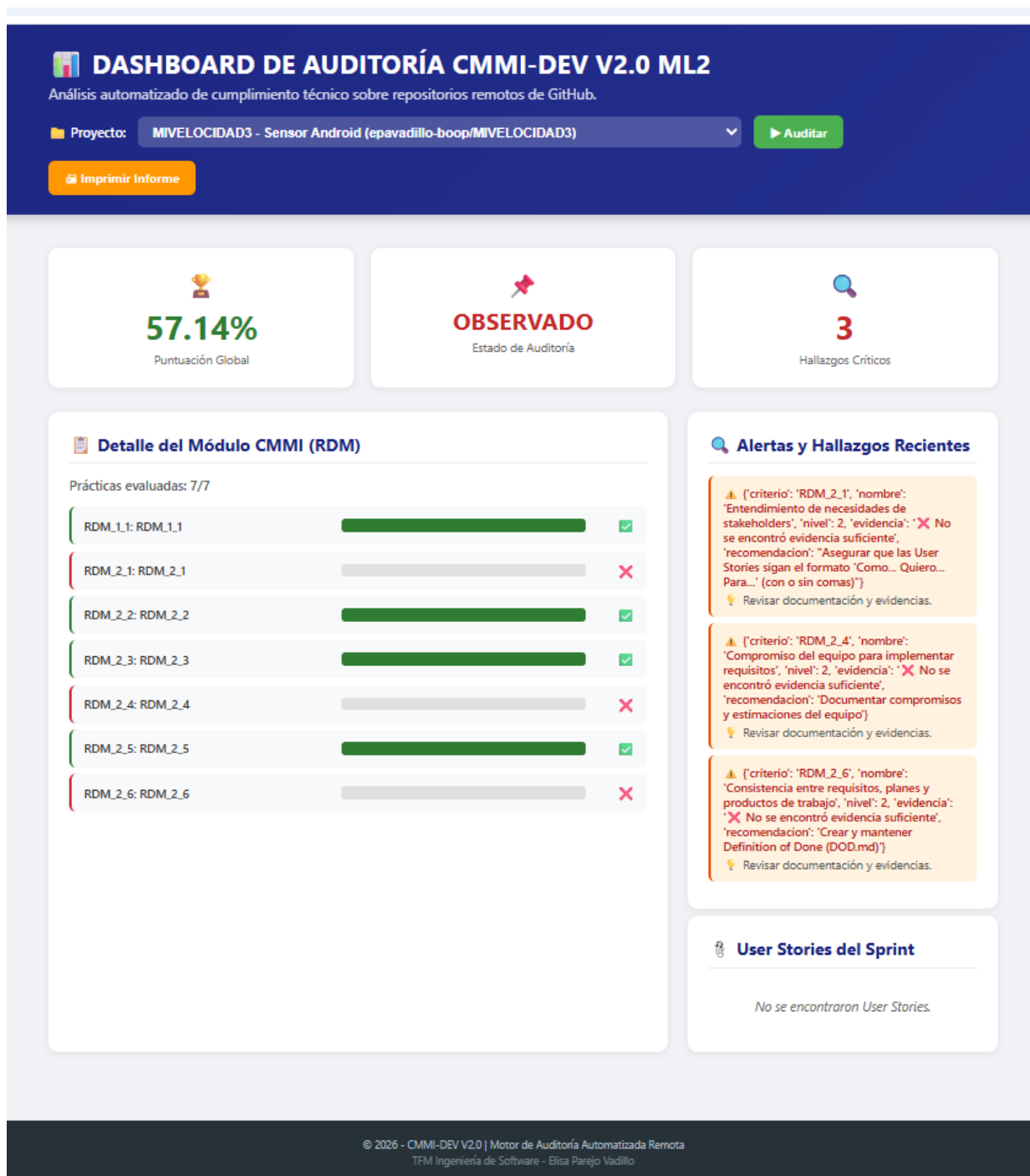


ILUSTRACIÓN 17. DASHBOARD CON CONFIGURACIÓN INICIAL PROYECTO MIVELOCIDAD3 (RDM)

Escenario 2: Configuración Multi-módulo (RDM + CM + PLAN)

Tras la refactorización, el sistema admite la inclusión de múltiples módulos mediante la modificación del manifiesto `estructura.txt`. La Ilustración 18 muestra la vista de detalle del módulo RDM en una configuración que incluye RDM, CM y PLAN:

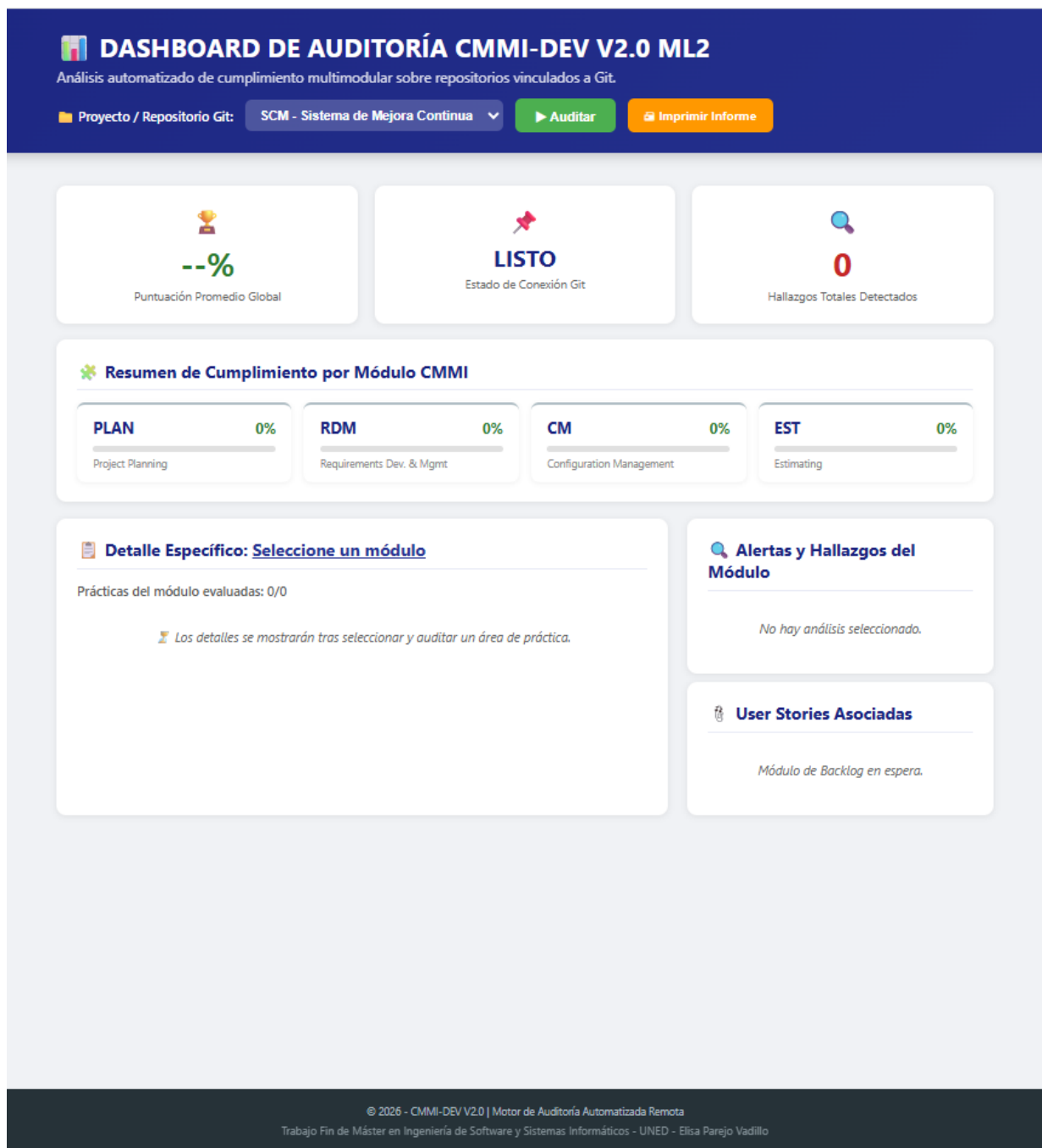


ILUSTRACIÓN 18. DASHBOARD PARA CONFIGURACIÓN MODULAR

Escenario 4: Dashboard Interactivo con Detalles Dinámicos

Finalmente, la Ilustración 19 muestra la interacción del usuario con el sistema, donde al seleccionar un módulo se despliegan los detalles de cumplimiento, criterios y hallazgos:

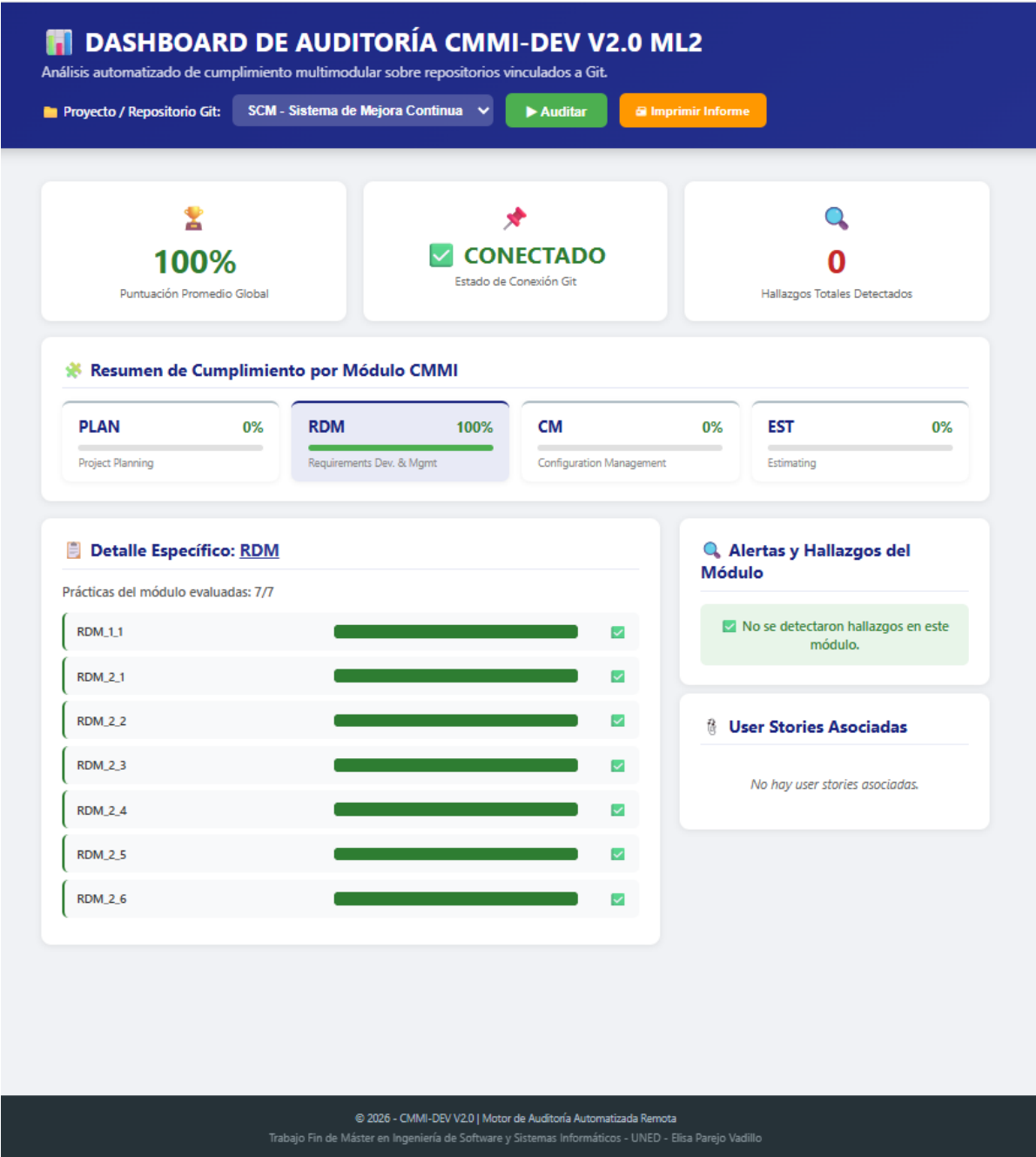


ILUSTRACIÓN 19. DASHBOARD VISTA DE DETALLE MÓDULO RDM EN CONFIGURACIÓN MULTI-MÓDULO

Escenario 4: Informe resultado de Auditoria impreso

La Ilustración 20 muestra el informe con el resultado de auditoría impreso.:

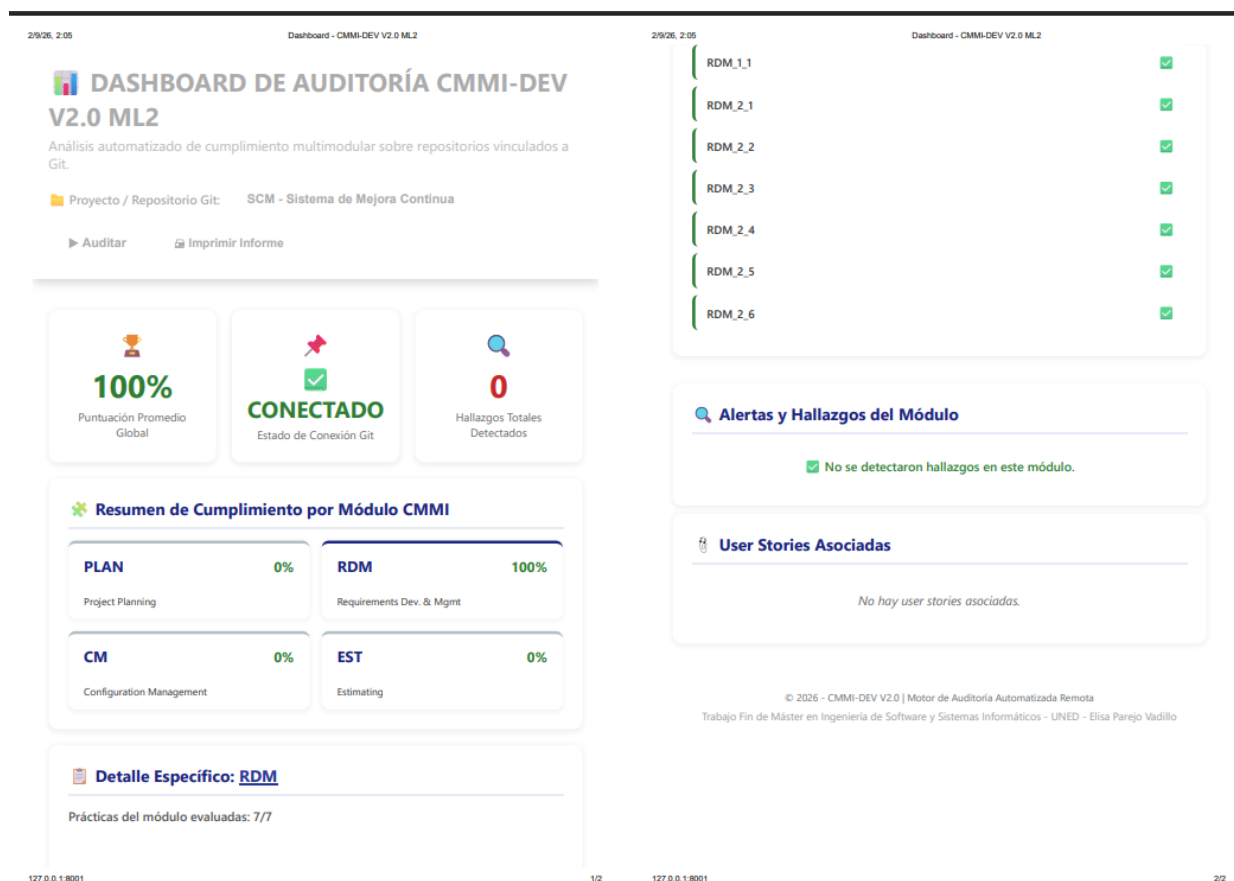


ILUSTRACIÓN 20. INFORME IMPRESO DESDE DASHBOARD RESULTADO EVALUACIÓN

5.5.5. Resultados de la Validación

La validación de la PoC arroja cuatro resultados:

1. Modularidad del sistema y Extensibilidad sin código. Añadir los módulos CM , PLAN y EST solo requirió modificar estructura.txt; no fue necesario escribir código nuevo. Los módulos se van implementando, añadiendo los módulos de reglas para las diferentes PA en `src/cmmi_rules/`.
2. Persistencia dinámica. El esquema de la base de datos se adapta a las nuevas áreas de práctica.
3. Interfaz reactiva. El dashboard se ajusta al número y tipo de módulos configurados.
4. Independencia del dominio. El motor aplica las reglas del módulo correspondiente sin conocer su naturaleza.

Estos resultados confirman la hipótesis del TFM: es posible construir un sistema de auditoría extensible que permita a una micropyme adoptar CMMI de forma progresiva y sin costes de desarrollo adicionales.

CAPITULO 6. Conclusiones y líneas futuras

6. Conclusiones y Líneas futuras

El presente capítulo resume los hallazgos alcanzados a lo largo de esta investigación, evalúa el grado de cumplimiento de los objetivos planteados. También propone las líneas de trabajo futuro que han surgido como resultado del trabajo realizado. Este trabajo ha abordado la intersección entre dos enfoques: la rigurosidad normativa del modelo CMMI-DEV V2.0 y la flexibilidad operativa de las metodologías ágiles, específicamente Scrum, en el contexto particular de las micropymes de consultoría tecnológica.

La pregunta de investigación que guio este trabajo fue ¿Cómo aplicar y automatizar el modelo CMMI DEV V2.0 de Nivel de Madurez 2 en una micropyme de consultoría industrial que ha transformado su propuesta de valor desde el desarrollo de proyectos basados en metodologías clásicas que para el desarrollo de la excelencia operacional hacia la consultoría tecnológica y el desarrollo de software ad-hoc, de forma que mejore la predictibilidad en sus plazos, costes y calidad, sin incurrir en una carga burocrática inviable para su tamaño y naturaleza?

A lo largo de los capítulos precedentes se ha dado respuesta a esta cuestión, demostrando que es posible integrar ambos enfoques. El desarrollo del modelo híbrido conceptual que se explica en el Capítulo 4, se construye un puente teórico entre CMMI DEV V2.0 y Scrum. Para la construcción de este modelo fueron de gran interés los estudios previos de Henríquez et al (Henriquez et al., 2021, 2022). Que desarrollan una matriz de relaciones entre artefactos Scrum y cumplimiento de prácticas del modelo CMM en su versión 1.3 y V2.0. Si bien el estudio se presenta desde una perspectiva de trazar las correspondencias entre los modelos mencionados, a efectos de este TFM, han supuesto una pieza fundamental para establecer los requisitos funcionales para el diseño de un motor de auditoría. Esta herramienta ha sido testada su despliegue en una PoC. El motor diseñado, es una herramienta de software que automatiza la auditoría CMMI y sirve como un mecanismo de control de calidad de los hitos del proceso de desarrollo.

Este TFM prueba que se reduce drásticamente la carga burocrática asociada tradicionalmente a la adopción de CMMI, con el uso de una herramienta de evaluación automática, como es el motor de auditoría.

6.1. Evaluación del Cumplimiento de los Objetivos

6.1.1. Cumplimiento Objetivo General

El objetivo general se ha alcanzado satisfactoriamente. Se ha diseñado un modelo híbrido que integra las prácticas del Área de Práctica RDM de CMMI-DEV V2.0 ML2 con los artefactos nativos de Scrum, y se ha implementado un motor de auditoría automatizada en Python (arquitectura FastAPI, persistencia SQLite, análisis mediante expresiones regulares) que escanea repositorios Git, valida la conformidad de los artefactos documentales y genera reportes de auditoría. La validación experimental sobre el repositorio piloto SCM-SISTEMA-MEJORA-CONTINUA. Para la verificación del funcionamiento de la PoC, se ha realizado la auditoría automática sobre un proyecto desarrollado a lo largo de este master para la asignatura de computación ubicua MIVELOCIDAD3. Aunque de cara a las pruebas del motor se modificaron los archivos documentales, para verificar que el proceso de auditoría era capaz evaluar adecuadamente los requisitos funcionales que se describen en el Capítulo 5 de este TFM. Los resultados del lanzamiento de las auditorías en la fase de *testing*, han demostrado la viabilidad de la solución. Como puede verse en las ilustraciones de la 16 a la 20 ambas incluidas.

6.1.2. Cumplimiento de Objetivos Específicos

Para cumplir este objetivo general, se establecieron 4 objetivos específicos que suponían un camino a seguir para llegar a la consecución del objetivo general

OE 1. Estudio del estado del arte: Este hito se cumple de forma exhaustiva en el Capítulo 3. Para ello se realizó un estudio histórico de la metodología CMMI, desde sus primeras estructuras rígidas del modelo SW CMMI, hasta la incorporación de la IA en la versión CMMI AIM. A lo largo de este estudio se analiza como la versión 2.0, marca un cambio de paradigma en el modelo al centra el rendimiento del negocio en el corazón del modelo. De la misma forma su flexibilidad le permite coexistir en entornos ágiles. La revisión de la literatura científica permitió detectar un vacío crítico, los estudios se centran en la convergencia Agile. CMMI desde una perspectiva metodológica pero no resuelve la burocracia que supone la realización de auditorías, ni reduce el coste de una implementación. Las suites ALM, no están concebidas para trabajar con equipos muy pequeños. Tienen una curva de aprendizaje muy alto y

resultan una carga económica inasumible por parte de una pyme o micropyme. Este hallazgo confirmó la necesidad de explorar vías para la automatización de las auditorías, mediante un análisis estático mediante un software de código abierto

OE 2. Diseño del modelo híbrido conceptual: Este objetivo se materializa en el Capítulo 4. La necesidad de agilizar los procesos en el entorno de desarrollo, ha llevado a las empresas a la adopción de modelos ágiles. Partiendo de esta tendencia empresarial, y a tenor de la revisión de la literatura relativa a la aplicación de las metodologías ágiles y su capacidad para atender a un modelo CMMI, se establece una matriz de comparación de las prácticas necesarias para la consecución de un nivel de madurez 2, y los artefactos scrum, para la ejecución de un proyecto ágil. En la matriz se establecen los artefactos nativos necesarios para el cumplimiento demandas operativas de las practicas. El *Product Backlog* asumió el rol de contenedor oficial de requisitos; las historias de usuario se establecieron como el vehículo para canalizar las necesidades de los *stakeholders*; y la Definición de Hecho (DoD) se consolidó como el contrato de calidad que garantiza la consistencia del software. Este objetivo incluyó la formalización de las restricciones sintácticas que permitieron al motor de software discriminar de manera matemática la conformidad documental.

OE 3. Desarrollo de una herramienta en Python: Para materializar la propuesta conceptual se construyó una herramienta de software bajo el *runtime* de Python 3.14. El *backend* se estructuró mediante una arquitectura de servicios API REST con FastAPI, empleando el ORM SQLAlchemy y una base de datos relacional embebida SQLite para garantizar un despliegue ligero y libre de costes de infraestructura. La lógica de control se imprimió mediante un enfoque modular, donde el motor clona de forma asíncrona el repositorio Git, mediante la librería GitPython. El motor separa la capa de presentación de la capa lógica, mediante un manifiesto (*estructura.txt*), para inyectar los metadatos en la interfaz web, mientras que las reglas de cumplimiento CMMI, son verificadas mediante expresiones regulares, que se encapsulan en el módulo independiente *rdm_scrum.py*. Para la escalabilidad del software, no hay que modificar el módulo basta con crear otro módulo, por ejemplo CM_Scrum.py, que contenga las reglas de control de esa PA

OE 4. Validación experimental: La herramienta demostró su capacidad de diagnóstico al someter a examen los dos casos de estudio configurados. En el proyecto piloto SCM-SISTEMA-MEJORA-CONTINUA, el motor certificó una puntuación del 100% global, almacenando de forma limpia en la base de datos SQLite la indexación de las historias de usuario (de la US-001 a la US-003), la presencia explícita de sus secciones de criterios de aceptación y las etiquetas de priorización reglamentarias, así como la validez estructural de las trazas en los *commits* y archivos de pruebas. Por el contrario, al auditar el repositorio MIVELOCIDAD3, el sistema bloqueó la conformidad del proyecto con una calificación del 57.14%, identificando con éxito 3 hallazgos críticos: la ausencia de la narrativa ágil formal ("Como/Quiero/Para") en el backlog, la falta de estimaciones y el incumplimiento de las directrices obligatorias de calidad dentro del archivo de la Definición de Hecho (DOD.md).

Objetivo Específico	Estado	Evidencia
OS1 Estudio del estado del arte	COMPLETADO	Capítulo 3: Análisis exhaustivo de la evolución de CMMI desde SW-CMM hasta CMMI v3.0 y AIM, comparativa de versiones y análisis de la literatura sobre integración CMMI-Scrum.
OS2. Diseño del modelo híbrido conceptual	COMPLETADO	Capítulo 4 y 5: Matriz de mapeo metodológico (Tabla 22), definición de patrones de sintaxis (expresiones regulares), análisis de brecha (<i>gap analysis</i>) completo, estructura documental definida.
OS3 Desarrollo de la herramienta Python	COMPLETADO	Capítulo 5.6: Implementación del motor FastAPI, módulos de reglas (rdm_scrum.py), persistencia SQLite, integración GitPython, <i>dashboard</i> reactivo. Demostración de modularidad mediante estructura.txt.
OS4 Validación experimental	COMPLETADO	Capítulo 5.6: Capturas de pantalla del <i>dashboard</i> (Ilustraciones 14-18), resultados de auditoría sobre repositorios piloto, demostración de extensibilidad con

inclusión de módulos CM y PLAN sin
modificar código.

TABLA 31. CUMPLIMIENTO DE OBJETIVOS ESPECÍFICOS

6.1.3. Contribuciones Principales del Trabajo

Contribución Metodológica

La principal contribución metodológica es la definición de un modelo híbrido conceptual que traduce las exigencias del Área de Práctica RDM de CMMI-DEV V2.0 ML2 al ecosistema nativo de Scrum. Este modelo:

- Define una correspondencia entre las prácticas de CMMI DEV V2.0 ML2y los artefactos ágiles (*Product Backlog, User Stories, Sprint Commitments, Definition of Done*).
- Establece expresiones regulares que permiten la verificación automatizada de la conformidad.
- Proporciona una guía para la implementación progresiva de CMMI CMMI DEV V2.0 ML2 en micropymes, evitando la sobrecarga burocrática tradicional.

Contribución Técnica

La contribución técnica es la más significativa es el desarrollo de un motor de auditoría automatizada con las siguientes características:

- **Extensibilidad sin código:** El sistema permite añadir nuevas áreas de práctica de CMMI (CM, PLAN, EST) mediante la simple modificación del archivo de configuración *estructura.txt*, sin necesidad de modificar el código fuente. La validación con los módulos CM y PLAN ha demostrado que la hipótesis de modularidad se cumple (véase Tabla 26).
- **Auditoría continua y objetiva:** El motor realiza verificaciones en tiempo de ejecución, elimina la subjetividad de las auditorías manuales y reduce de forma considerable el esfuerzo de generación de evidencias documentales.

- **Trazabilidad bidireccional garantizada:** Mediante el análisis de las historias de usuario y sus criterios de aceptación, el sistema asegura que cada requisito esté vinculado a su implementación y verificación, cumpliendo con RDM 2.4 y RDM 2.6 de forma automatizada.
- **Bajo coste y autocontenido:** Al estar desarrollado con tecnologías de código abierto y ejecutarse sobre SQLite, la herramienta no requiere inversión en infraestructura ni licencias, haciéndola viable para micropymes.

Contribución al Conocimiento en Ingeniería de Software

Este trabajo demuestra empíricamente, que es viable la integración de los modelos de mejora de procesos (CMMI DEV V2.0 ML2) y los marcos ágiles (Scrum). Se puede concluir que:

- La tensión entre ambos enfoques puede resolverse mediante la automatización, reduciendo la carga burocrática respetando el control estricto de los procesos.
- La modularidad arquitectónica es clave para la adopción escalonada de CMMI DEV V2.0 ML2, permitiendo a las organizaciones empezar con un área de práctica e ir añadiendo otras progresivamente.
- Es posible abordar los distintos niveles de madurez con el enfoque de la arquitectura modular, sin modificación del código. Esta extensibilidad es posible, traduciendo las necesidades normativas a los ficheros que soportan las reglas CMMI DEV V2.0 ML2.
- El análisis sintáctico mediante expresiones regulares es un mecanismo eficaz y de bajo coste para la verificación de conformidad documental en repositorios ágiles, como alternativa al procesamiento semántico más costoso.

6.2. Lecciones aprendidas

Durante el desarrollo de esta investigación se han extraído diversas lecciones que resultan relevantes para futuros trabajos y para la práctica profesional:

6.2.1. Sobre el proceso de adopción de CMMI en micropymes.

- **El diagnóstico inicial (*gap analysis*) es crítico:** Sitúa a la organización frente a su espejo, y permite conocer sus fortalezas y debilidades. El diagnóstico permite establecer una hoja de ruta para la consecución de la certificación
- **La documentación como barrera y como oportunidad:** El principal desafío identificado en la evaluación es la formalización documental, muchas prácticas se ejecutan, pero no existe registro, y si existe resulta laborioso auditar su existencia, más allá de la integridad del propio documento, que no es objeto de la evaluación. Este hallazgo justifica la necesidad de herramientas que automaticen la generación y verificación de evidencias.
- **La extensibilidad de la herramienta es su principal valor:** La capacidad de añadir nuevos módulos (CM, PLAN) sin modificar el código central es lo que hace viable la adopción progresiva de CMMI.

6.2.2. Sobre el desarrollo de la PoC

- **La modularidad ha precisado de una refactorización:** La transformación del sistema monolítico (solo RDM) a una arquitectura dirigida por datos con (estructura.txt) requirió intervenir en todas las capas del software (véase Tabla 26). Sin embargo, este esfuerzo permite la extensión sin nuevo código.
- **Las expresiones regulares han demostrado ser útiles, pero presentan limitaciones:** El análisis sintáctico ha demostrado ser eficaz para la validación documental de los artefactos Scrum que permiten identificar un patrón ("Como/Quiero/Para"), sus condiciones de aceptación y reglas como DoD. Sin embargo, las estructuras sintácticas no son suficientes para evaluar la calidad real del contenido de los artefactos, se requiere análisis semántico para su evaluación
- **La interfaz de usuario guiada por datos (Data-Driven UI) es clave para la usabilidad:** El rediseño del dashboard para que descubra dinámicamente los módulos disponibles en la base de datos es determinante para la usabilidad de

la herramienta, y ha servido para la demostrar la modularidad de la arquitectura del software.

6.3. Limitaciones del Estudio

Las principales limitaciones que condicionan la generalización de los resultados obtenidos son las siguientes:

1. **Área de práctica acotada:** La PoC se ha centrado exclusivamente en el Área de Práctica RDM (Desarrollo y Gestión de Requisitos) del nivel de madurez 2. Aunque la arquitectura es extensible, la validación de otras áreas (como CM, PLAN, EST, PQA, SAM) requeriría desarrollar los módulos de reglas correspondientes y adaptar el análisis a sus prácticas específicas.
2. **Análisis sintáctico exclusivamente:** El motor realiza un análisis basado en la presencia de patrones textuales, no en el significado semántico del contenido. Esto implica que no puede evaluar la calidad intrínseca de las historias de usuario o la pertinencia de los criterios de aceptación.
3. **Dependencia del formato Markdown:** La herramienta está optimizada para repositorios que utilizan Markdown como estándar documental. No se ha probado en entornos con otros formatos (Word, Excel, herramientas ALM propietarias).
4. **Entorno de validación acotado:** La validación experimental se ha realizado sobre un repositorio piloto de una única organización (el proyecto SCM-SISTEMA-MEJORA-CONTINUA). Aunque se han probado dos proyectos (SCM y MIVELOCIDAD3), no se ha realizado una validación con múltiples equipos o en diferentes contextos organizacionales.
5. **Evaluación informal del *gap analysis*:** El diagnóstico inicial se realizó mediante una preevaluación interna (no oficial), no mediante una evaluación SCAMPI formal. Esto condiciona la validez de los resultados obtenidos en el análisis de brecha.

6.4. Líneas de Trabajo Futuro

Las líneas de trabajo futuro se estructuran en tres niveles: extensión metodológica, mejora técnica y nuevos horizontes de aplicación.

6.4.1. Extensión a Otras Áreas de Práctica de CMMI

La arquitectura modular desarrollada permite la extensión progresiva a otras áreas de práctica del nivel de madurez 2, y eventualmente a niveles superiores:

- **Corto plazo:** Desarrollo de módulos para CM (Gestión de Configuración), PLAN (Planificación de Proyectos) y EST (Estimación). Ya se ha demostrado la viabilidad técnica al incluir CM y PLAN sin modificar el código central. El trabajo pendiente consiste en definir las expresiones regulares y los criterios de validación para estas áreas.
- **Medio plazo:** Extensión a PQA (Aseguramiento de Calidad del Proceso) y SAM (Gestión de Proveedores). Ambas áreas fueron identificadas como críticas en el *gap analysis* (estado NOT SATISFIED) y su automatización abordaría las brechas más severas detectadas.
- **Largo plazo:** Incorporación de áreas de nivel superior (ML3, ML4, ML5) como RSK (Gestión de Riesgos), MPM (Medición de Rendimiento), CAR (Análisis de Causa Raíz) o PCM (Gestión de Cambios de Proceso).

6.4.2. Mejoras Técnicas y Arquitectónicas

- **Integración semántica mediante LLMs:** Incorporar procesamiento de lenguaje natural (PLN) mediante modelos de lenguaje (LLMs) para evaluar no solo la presencia de patrones sintácticos, sino la calidad semántica de las historias de usuario, la coherencia de los criterios de aceptación y la validez completa de los requisitos. Esto permitiría superar la principal limitación identificada.

- **Notificaciones y alertas automáticas:** Implementar un sistema de notificaciones (vía correo electrónico, *Slack* o integración con herramientas de gestión de proyectos) que informe al equipo de los hallazgos de la auditoría en tiempo real.
- **Generación automática de evidencias:** Desarrollar un módulo que, a partir de los resultados de la auditoría, genere automáticamente las evidencias documentales requeridas para una evaluación formal de CMMI (reportes de trazabilidad, matrices de requisitos)
- **Dashboard avanzado:** Mejorar la interfaz con gráficos de evolución temporal (seguimiento del cumplimiento a lo largo de los *sprints*), comparativa entre proyectos.

CAPITULO 7. Bibliografía y Referencias.

7. Bibliografía y referencias

467628891-Booklet-CMMI-V2-0-Model-At-A-Glance-Digital-Spanish-2019-06-28-pdf. (s. f.).

Alves, L., Souza, G., Ribeiro, P., & Machado, R.-J. (2021). Longevity of risks in software development projects: A comparative analysis with an academic environment. En *Procedia Computer Science* (Vol. 181, pp. 827-834).

<https://doi.org/10.1016/j.procs.2021.01.236>

Arjun, K. P., & Smitha, V. (2025). *Smart Factories in the Era of Industry 4.0: Convergence of IoT and Cyber-Physical Systems for Intelligent Automation, Connectivity, and Resilient Manufacturing*. 261-270. https://doi.org/10.2991/978-94-6463-914-8_17

Aymerich Fuentes, B., & Jenkins, M. (2022). Use of CMMI-DEV 2.0 for continuous process improvement in an agile organization: A case study. *2022 17th Iberian Conference on Information Systems and Technologies (CISTI)*, 1-6.

<https://doi.org/10.23919/CISTI54924.2022.9820443>

Balla, K., MiaoMiao, T., Mowat, P., Rasking, M., Shang, C., & van Veenendaal, E. (2020). *Changes in CMMI 2.0 and how they can affect TMMi*.

Bradac Hojnik, B., Hušek, I., Smit, S., De la Chevsnier, M.-A., & Mixchke, J. (2023). Small and Medium-Sized Enterprises in the Digital Age: Understanding Characteristics and Essential Demands. *Information*, 14, 606. <https://doi.org/10.3390/info14110606>

Carnegie Mellon University. (2010). *CMMI® para Desarrollo, Versión 1.3*.

Chrissis, M. B., Konrad, M., & Shrum, S. (2011). *CMMI for Development: Guidelines for Process Integration and Product Improvement*. Pearson Education.

CMMI Institute. (s. f.-a). *CMMI Institute—Certifications*. Recuperado 25 de agosto de 2026, de <https://cmmiinstitute.com/learning/certifications>

- CMMI Institute. (s. f.-b). *CMMI Institute—¿Qué es CMMI?* Recuperado 9 de agosto de 2026, de <https://cmmiinstitute.com/cmmi/intro-3>
- CMMI Institute. (2018). *CMMI Development V2.0*.
- CMMI Institute. (2019). *Booklet-cmmi-v2-0-model-at-a-glance-digital-spanish-2019-06-28-pdf*.
- CMMI Institute. (2026a). *CMMI Acquisition Handbook*.
- CMMI Institute. (2026b). *CMMI Adoption Guidance*. <https://cmmiinstitute.com/resource-files/public/v2-0-materials/cmmi-v2-0-adoption-and-transition-guide>
- Cukier, J. J. (2008). *Problemas de las pymes en el nivel 2 de madurez. Una muestra sesgada*. (2).
- Def-guide-to-sqlite/2829.pdf at master · Apress/def-guide-to-sqlite*. (s. f.). GitHub. Recuperado 1 de septiembre de 2026, de <https://github.com/Apress/def-guide-to-sqlite/blob/master/2829.pdf>
- FunSoft2020, S. (2020, mayo 10). *Understanding CMMI – Blog SWA – UC3M*. <https://blogswa.sel.inf.uc3m.es/?p=5989#comment-973>
- Giordano, M., Hieronimus, S., Smit, S., De la Chevasnerie, M.-A., & Mischke, J. (2024, enero 16). *Accelerating European economic competitiveness | McKinsey*. <https://www.mckinsey.com/mgi/our-research/accelerating-europe-competitiveness-for-a-new-era>
- Global Quality Services. (2026, febrero 3). *CMMI 2.0 Certification Process Guide*. <https://iso-certifications.id/cmmi-2-0-certification-indonesia/>
- Happonen, A., Santti, U., Auvinen, H., Räsänen, T., & Eskelinen, T. (2020). Digital age business model innovation for sustainability in University Industry Collaboration Model. En *E3S Web of Conferences* (Vol. 211). <https://doi.org/10.1051/e3sconf/202021104005>

Henriquez, V., Calvo-Manzano, J. A., Moreno, A. M., & San Feliu, T. (2022). Agile-CMMI V2.0 alignment: Bringing to light the agile artifacts pointed out by CMMI. *Computer Standards & Interfaces*, 82, 103610. <https://doi.org/10.1016/j.csi.2021.103610>

Henriquez, V., Moreno, A. M., Calvo-Manzano, J. A., & San Feliu, T. (2021). *Agile-CMMI Alignment: CMMI V2.0 Contributions and To-dos for Organizations*. arXiv. <https://doi.org/10.48550/arXiv.2110.00399>

ISACA. (2021a, junio 7). *CMMI by the Numbers | Infographic*. ISACA. <https://www.isaca.org/resources/infographics/cmmi-by-the-numbers>

ISACA. (2021b, junio 7). *Take Organizational Performance to the Next Level with CMMI® V2.0 | White Paper*. ISACA. <https://www.isaca.org/resources/infographics/take-organizational-performance-to-the-next-level-with-cmmi>

ISACA. (2025, diciembre 17). *CMMI: How to Purchase the CMMI Model?* <https://support.isaca.org/s/article/How-do-I-access-the-CMMI-V2-0-model-1598331746260>

ISO 25010. (s. f.). Recuperado 5 de agosto de 2026, de <https://iso25000.com/index.php/normas-iso-25000/iso-25010>

ISO IEC 25010 2011 Systems and Software Quality Requirements and Evaluation SQuaRE Quality Model.pdf. (s. f.). Slideshare. Recuperado 1 de septiembre de 2026, de <https://www.slideshare.net/slideshow/iso-iec-25010-2011-systems-and-software-quality-requirements-and-evaluation-square-quality-modelpdf/256282123>

Kitchenham, B., Dybå, T., & Jorgensen, M. (2004). *Evidence-based software engineering* (p. 281). <https://doi.org/10.1109/ICSE.2004.1317449>

Kruchten, P. (s. f.). *Planos Architect'onicos: El Modelo de "4+1" Vistas de la Arquitectura del Software*.

- Libo, D. (2025). The Role of Digital Transformation in Enhancing Employee Competencies: A CMMI-Mediated Approach Towards Sustainable Organizational Development. *Journal of Information Systems Engineering and Management*, 10, 32-43.
<https://doi.org/10.52783/jisem.v10i28s.4302>
- McGraw, G. (2006). *Software Security: Building Security in*. Addison-Wesley Professional.
- Niazi, M., Saeed, A., Alshayeb, M., Mahmood, S., & Zafar, S. (2020). A Maturity Model for Secure Requirements Engineering. *Computers & Security*, 95, 101852.
<https://doi.org/10.1016/j.cose.2020.101852>
- Nielsen, J. (with Internet Archive). (1993). *Usability engineering*. Boston : Academic Press.
<http://archive.org/details/usabilityenginee0000niel>
- Palomeque-Jaramillo, J., Dimitrakaki, I., & Solano, J. (2024). *Innovación y Competitividad en la Transformación Digital*. https://doi.org/10.37572/EdArt_031224420
- Pussella, H. D. M. M., & Bandara, A. (2018). Exploring the Challenges in Transitioning from Traditional Project Management to Agile Project Management. *Peradeniya Management Review*, 1, 17-49. <https://doi.org/10.4038/pmr.v1i1.25>
- Ramzy, F. (2026, julio 31). (20) *Clean Code—Chapter 9: Unit Tests* | *LinkedIn*.
<https://www.linkedin.com/pulse/clean-code-chapter-9-unit-tests-fady-ramzy-ozh0f/>
- Reifer, D. J. (2000). The CMMI: It's formidable. *Journal of Systems and Software*, 50(2), 97-98. [https://doi.org/10.1016/S0164-1212\(99\)00119-3](https://doi.org/10.1016/S0164-1212(99)00119-3)
- Response Time Limits: Article by Jakob Nielsen*. (s. f.). Nielsen Norman Group. Recuperado 1 de septiembre de 2026, de <https://www.nngroup.com/articles/response-times-3-important-limits/>
- Reynoso, J. M. G., Andrade, E. L. M., & Díaz, J. E. M. (2010). *Utilizando el Modelo de Calidad de McCall y el Estándar ISO-9126 para la Evaluación de la Calidad de Sistemas de Información por los Usuarios*.

Robinson, J. (2026). *How do organizations measure the ROI of implementing CMMI, and what time frame is realistic for expecting visible results?*

[https://flevy.com/topic/cmmi/question/maximizing-cmmi-roi-measuring-impact-realistic-](https://flevy.com/topic/cmmi/question/maximizing-cmmi-roi-measuring-impact-realistic-timeframes?srltid=AfmBOopbFUzhQ3eFkgEGmEo7H19_HXXiCtjFUmyahh65VGQ9824TaEvM)

[timeframes?srltid=AfmBOopbFUzhQ3eFkgEGmEo7H19_HXXiCtjFUmyahh65VGQ9](https://flevy.com/topic/cmmi/question/maximizing-cmmi-roi-measuring-impact-realistic-timeframes?srltid=AfmBOopbFUzhQ3eFkgEGmEo7H19_HXXiCtjFUmyahh65VGQ9824TaEvM)

[824TaEvM](https://flevy.com/topic/cmmi/question/maximizing-cmmi-roi-measuring-impact-realistic-timeframes?srltid=AfmBOopbFUzhQ3eFkgEGmEo7H19_HXXiCtjFUmyahh65VGQ9824TaEvM)

The Agile Manifesto Authors. (2001). *Agile Manifesto for Software Development* | *Agile Alliance*. <https://agilealliance.org/agile101/the-agile-manifesto/>

UI, G. C. U. |. (2024, junio 14). What Are Semantic Colours In a Design System? *Bootcamp*.

[https://medium.com/design-bootcamp/what-are-semantic-colours-in-a-design-system-](https://medium.com/design-bootcamp/what-are-semantic-colours-in-a-design-system-993e1f7362da)

[993e1f7362da](https://medium.com/design-bootcamp/what-are-semantic-colours-in-a-design-system-993e1f7362da)

Valera, F., & Solutions, V. (2025, mayo 14). CMMI: Capability Maturity Model Integration (Complete Guide). *Visure Solutions*. <https://visuresolutions.com/alm-guide/capability-maturity-model-integration-cmmi/>

van Veenendaal, E. (2026). *TMMi Framework model V2.0*. <https://www.tmmi.org>

Veenendaal, E., Balla, K., & Cannegieter, J. J. (2023). *THE NEW MDD AND TAMAR; A STUDY ON HOW MDD, THE NEW CMMI ASSESSMENT METHOD, CAN AFFECT TMMI ASSESSMENTS*.

Villar, E., Martín Toral, I., Calvo, I., Barambones, O., & Fernández-Bustamante, P. (2024).

Architectures for Industrial AIoT Applications. *Sensors*, 24(15), 4929.

<https://doi.org/10.3390/s24154929>

Wood, P. B., & Vickers, D. (2018). Anticipated impact of the capability maturity model integration (CMMI®) V2.0 on aerospace systems safety and security. *2018 IEEE Aerospace Conference*, 1-11.

<https://doi.org/10.1109/AERO.2018.8396579>

LISTADOS

Listado de Siglas

- **ACQ (Acquisition):** Adquisiciones (Referido a la antigua constelación CMMI-ACQ).
- **AIM (Artificial Intelligence Maturity):** Madurez en Inteligencia Artificial (Extensión del modelo CMMI v3.0).
- **ALM (Application Lifecycle Management):** Gestión del Ciclo de Vida de las Aplicaciones.
- **API (Application Programming Interface):** Interfaz de Programación de Aplicaciones.
- **ATM (Appraisal Team Member):** Miembro del Equipo de Evaluación (CMMI).
- **BSC (Business Sustaining Capability):** Habilitador o Capacidad de Sustento del Negocio (Categoría en CMMI V2.0).
- **CA (Capability Area):** Área de Capacidad (CMMI V2.0).
- **CAR (Causal Analysis and Resolution):** Análisis de Causa Raíz y Resolución (Área de Práctica de CMMI).
- **CI (Configuration Item / Continuous Integration):** Ítem de Configuración / Integración Continua (Según el contexto del documento).
- **CD (Continuous Deployment):** Despliegue Continuo.
- **CL (Capability Level):** Nivel de Capacidad (CMMI).
- **CM (Configuration Management):** Gestión de la Configuración (Área de Práctica de CMMI).
- **CMM (Capability Maturity Model):** Modelo de Madurez de Capacidades.
- **CMMI (Capability Maturity Model Integration):** Integración de Modelos de Madurez de Capacidades.
- **CRIT-:** Prefijo o Token identificador empleado en la PoC para los Criterios de Aceptación.
- **CSS3 (Cascading Style Sheets 3):** Hojas de Estilo en Cascada, Versión 3.
- **CU (Caso de Uso):** Elemento de modelado UML empleado en la Vista de Escenarios.
- **DAR (Decision Analysis and Resolution):** Análisis y Resolución de Decisiones (Área de Práctica de CMMI).

- **DEV (Development):** Desarrollo (Referido a la vista o constelación CMMI-DEV).
- **DoD (Definition of Done):** Definición de Hecho o Terminado (Marco Scrum).
- **DOD-:** Prefijo o Token identificador empleado en la PoC para las reglas de la Definición de Hecho.
- **EDP (Engineering and Developing Products):** Ingeniería y Desarrollo de Productos (Área de Capacidad en CMMI).
- **ENQ (Ensuring Quality):** Aseguramiento de la Calidad (Área de Capacidad en CMMI).
- **EST (Estimating):** Estimación (Área de Práctica de CMMI).
- **FI (Fully Implemented):** Totalmente Implementada (Criterio de calificación oficial de ISACA).
- **GG (Generic Goal):** Meta Genérica (Modelos CMMI V1.3 e inferiores).
- **GOV (Governance):** Gobernanza (Área de Práctica de CMMI V2.0).
- **GP (Generic Practice):** Práctica Genérica (Modelos CMMI V1.3 e inferiores).
- **HTML5 (HyperText Markup Language 5):** Lenguaje de Marcado de Hipertexto, Versión 5.
- **HTTP (Hypertext Transfer Protocol):** Protocolo de Transferencia de Hipertexto.
- **HTTPS (Hypertext Transfer Protocol Secure):** Protocolo Seguro de Transferencia de Hipertexto.
- **II (Implementation Infrastructure):** Infraestructura de Implementación (Área de Práctica de CMMI V2.0).
- **IIoT (Industrial Internet of Things):** Internet Industrial de las Cosas.
- **IMP (Improving Performance):** Mejora del Rendimiento (Área de Capacidad en CMMI).
- **I/O (Input / Output):** Entrada / Salida.
- **ISACA (Information Systems Audit and Control Association):** Asociación de Auditoría y Control de Sistemas de Información (Entidad propietaria de CMMI).
- **ISO (International Organization for Standardization):** Organización Internacional de Normalización.
- **JSON (JavaScript Object Notation):** Notación de Objetos JavaScript (Formato universal de intercambio de datos).
- **KPI (Key Performance Indicator):** Indicador Clave de Rendimiento.

- **LI (Largely Implemented):** Mayoritariamente Implementada (Criterio de calificación oficial de ISACA).
- **LLM (Large Language Model):** Modelo de Lenguaje de Gran Tamaña (Inteligencia Artificial).
- **MBR (Managing Business Resilience):** Gestión de la Resiliencia del Negocio (Área de Capacidad en CMMI).
- **MC (Monitor and Control):** Monitorización y Control (Área de Práctica de CMMI).
- **MDD (Method Definition Document):** Documento de Definición del Método de Evaluación de CMMI.
- **MIVELOCIDAD3:** Nombre identificador de uno de los repositorios Git utilizados en las pruebas piloto de la PoC.
- **ML (Maturity Level):** Nivel de Madurez (CMMI).
- **ML2 (Maturity Level 2):** Nivel de Madurez 2.
- **MPM (Managing Performance and Measurement):** Gestión del Rendimiento y Medición (Área de Práctica de CMMI).
- **MWF (Managing the Workforce):** Gestión de la Fuerza de Trabajo / Personal (Área de Capacidad en CMMI).
- **NI (Not Implemented):** No Implementada (Criterio de calificación oficial de ISACA).
- **ORM (Object-Relational Mapping):** Mapeo Objeto-Relational.
- **OS1 / OS2 / OS3 / OS4:** Objetivos Específicos del Trabajo Fin de Máster.
- **OT (Organizational Training):** Capacitación Organizacional (Área de Práctica de CMMI).
- **PA (Practice Area / Process Area):** Área de Práctica (en CMMI V2.0) / Área de Proceso (en CMMI V1.3).
- **PAD (Process Asset Development):** Desarrollo de Activos de Procesos (Área de Práctica de CMMI).
- **PCM (Process Capability Management):** Gestión de la Capacidad de Procesos (Área de Práctica de CMMI).
- **PI (Partially Implemented / Product Integration):** Parcialmente Implementada / Integración del Producto (Según el contexto del documento).
- **PLAN (Project Planning):** Planificación de Proyectos (Área de Práctica de CMMI).

- **PLN / NLP (Procesamiento de Lenguaje Natural):** Procesamiento de Lenguaje Natural.
- **PMW (Planning and Managing Work):** Planificación y Gestión del Trabajo (Área de Capacidad en CMMI).
- **PO (Product Owner):** Propietario del Producto (Marco Scrum).
- **PoC (Proof of Concept):** Prueba de Concepto.
- **PQA (Process and Product Quality Assurance):** Aseguramiento de la Calidad del Proceso y del Producto (Área de Práctica de CMMI).
- **PR (Peer Reviews):** Revisiones por Pares (Área de Práctica de CMMI V2.0).
- **PYME:** Pequeña y Mediana Empresa.
- **RDM (Requirements Development and Management):** Desarrollo y Gestión de Requisitos (Área de Práctica de CMMI V2.0).
- **Regex (Regular Expression):** Expresión Regular. Patrón de búsqueda
- **REST (Representational State Transfer):** Transferencia de Estado Representacional (Estilo de arquitectura de software para APIs).
- **RFC (Request for Comments / Request for Change):** Solicitud de Cambio (Según el contexto del documento).
- **RNF-01 / RNF-02 / ...:** Códigos identificadores de los Requisitos No Funcionales.
- **RF-01 / RF-02 / ...:** Códigos identificadores de los Requisitos Funcionales.
- **ROI (Return on Investment):** Retorno de la Inversión.
- **RSK (Risk Management):** Gestión de Riesgos (Área de Práctica de CMMI).
- **SAM (Supplier Agreement Management):** Gestión de Acuerdos con Proveedores (Área de Práctica de CMMI).
- **SCAMPI (Standard CMMI Appraisal Method for Process Improvement):** Método de Evaluación Estándar de CMMI para la Mejora de Procesos (Modelos v1.3 e inferiores).
- **SCM (Software Configuration Management):** Gestión de la Configuración de Software.
- **SE (Systems Engineering):** Ingeniería de Sistemas.
- **SG (Specific Goal):** Meta Específica (Modelos CMMI V1.3 e inferiores).
- **SI (Supporting Implementation):** Soporte a la Implementación (Área de Capacidad en CMMI).

- **SLA (Service Level Agreement):** Acuerdo de Nivel de Servicio.
- **SM (Scrum Master):** Facilitador u orquestador del marco de trabajo Scrum.
- **SP (Specific Practice):** Práctica Específica (Modelos CMMI V1.3 e inferiores).
- **SPI (Software Process Improvement):** Mejora de Procesos de Software.
- **SVC (Services):** Servicios (Referido a la antigua constelación CMMI-SVC).
- **SW-CMM (Software CMM):** Modelo de Madurez de Capacidades específico para Software.
- **TDD (Test-Driven Development):** Desarrollo Guiado por Pruebas.
- **TFM:** Trabajo Fin de Máster.
- **TMMi (Test Maturity Model integration):** Modelo de Integración de Madurez de Pruebas.
- **TS (Technical Solution):** Solución Técnica (Área de Práctica de CMMI).
- **UI (User Interface):** Interfaz de Usuario.
- **UML (Unified Modeling Language):** Lenguaje de Modelado Unificado.
- **UNED:** Universidad Nacional de Educación a Distancia.
- **US- / User Story:** Historia de Usuario.
- **VCS (Version Control System):** Sistema de Control de Versiones.
- **VV (Verification and Validation):** Verificación y Validación (Área de Práctica de CMMI V2.0).
- **WBS (Work Breakdown Structure):** Estructura de Desglose del Trabajo (EDT).

ANEXOS

Anexo I. Diseño de la pre-evaluación interna:

El objetivo del diseño de la pre-evaluación interna, es preparar a la empresa para una auditoria oficial. Si bien la auditoria oficial no puede ser realizada por el Scrum Master, para que no existan un conflicto de calidad en las respuestas. Para la realización de este TFM, se ha diseñado la pre-evaluación para que el Scrum Master, sea quien la realice de cara a su formación y para que sirva de autodiagnóstico.

Para el diseño de la pre-evaluación interna se crean dos formularios, uno de entrevistas donde el scrum máster responde a un cuestionario sobre cual es el procedimiento operativo para el cumplimiento de prácticas y cuáles serían las evidencias que soportan sus afirmaciones, y otro formulario donde se registran las evidencias verificadas para cada práctica

Formulario de entrevistas:

Para el seguimiento de entrevistas se crea un libro de Excel que contiene. Una hoja por cada categoría. Dentro de la hoja se encuentra el listado de todas las prácticas asociada a esa categoría para el modelo CMMI DEV V2.0 del nivel de Madurez 2. Que se encuentran recogidas en la Tabla 16. Listado de prácticas por PA's CMMI DEV V2.0 ML2 de este TFM

1. Categoría
2. Área de Capacidad
3. Practice Area
4. ID Práctica
5. Pregunta del Auditor en Vivo
6. Respuesta Clave del Scrum Master (Qué decir)
7. Evidencia Digital a Proyectar (Qué mostrar)

Se muestra a continuación el resultado de la evaluación para el área de Manging.

Resultados_Evaluacion_entrevistas_CMMI_DEV_V20_ML2_GUION ENTREVISTA MANAGING

Categoría	Área de Capacidad	Practice Area	ID Práctica	Pregunta del Auditor en Vivo	Respuesta Scrum Master	Evidencia Digital
MANAGING	Planning and Managing Work	EST	EST 1.1	¿Dónde guardan la primera versión de las estimaciones del proyecto o estimaciones macro de alto nivel?	Registramos las estimaciones iniciales y de alto nivel en nuestra propuesta técnica (archivada en Drive) para mantener una línea base histórica y de referencia antes de iniciar el desglose detallado.	Drive: Documento inicial aprobado con las estimaciones macro de arranque.
MANAGING	Planning and Managing Work	EST	EST 2.1	¿Cómo aseguran que estiman el alcance completo del trabajo basándose en criterios definidos y que no olvidan requerimientos?	No estimamos requerimientos sueltos. Desglosamos el alcance completo en un Product Backlog estructurado en Épicas e Historias de Usuario dentro de nuestra herramienta de gestión operativa.	El Backlog del proyecto ordenado jerárquicamente con el alcance completo mapeado.
MANAGING	Planning and Managing Work	EST	EST 2.2	¿De qué manera estiman las características, tamaño o complejidad de los productos de trabajo y las tareas técnicas?	El equipo técnico realiza una sesión de estimación relativa (como Planning Poker). Evaluamos el tamaño y la complejidad asignando Story Points a cada una de las Historias de Usuario.	Creamos un documento excel con Historias de usuario con sus Story Points cargados de forma visible en las tarjetas.
MANAGING	Planning and Managing Work	EST	EST 2.3	¿Cómo transforman esas estimaciones de tamaño en semanas, horas hombre y presupuesto, y dónde dejan registro de su justificación?	Tomamos los Story Points y, utilizando la velocidad histórica del equipo, calculamos el esfuerzo, la duración y el coste. Dejamos registrados en el documento los supuestos técnicos y restricciones que lo justifican.	Excel de Estimación de Históricos. Se alimenta en base a la experiencia del equipo.
MANAGING	Planning and Managing Work	PLAN	PLAN 1.1	Al arrancar operativamente el proyecto, ¿cómo definen y desglosan la lista de tareas iniciales y asignan al personal disponible?	Creamos una lista inicial de tareas operativas para arrancar el trabajo y las asignamos formalmente a los miembros del equipo técnico según su disponibilidad y especialidad.	Excel: Lista de tareas iniciales cargadas y asignadas a los usuarios del equipo.
MANAGING	Planning and Managing Work	PLAN	PLAN 1.2	¿Cómo se asignan las tareas?	en la reunión del lanzamiento del proyecto se identifican tareas y responsables en función del área de especialización	Excel: Lista de tareas iniciales cargadas y asignadas a los usuarios del equipo.
MANAGING	Planning and Managing Work	PLAN	PLAN 2.1	¿Qué enfoque o ciclo de vida técnico seleccionaron para la ejecución de este proyecto y en dónde se encuentra documentado?	Analizamos las características del proyecto y seleccionamos el tipo de enfoque en función de esto. Este enfoque se justifica y documenta formalmente en nuestro Plan de Proyecto.	Drive -Plan de Proyecto: Plan de Proyecto donde se describe y justifica la selección del ciclo de vida elegido.
MANAGING	Planning and Managing Work	PLAN	PLAN 2.2	Si el proyecto requiere conocimientos o habilidades específicas, ¿cómo planifican asegurar que el equipo cuente con las capacidades necesarias?	Identificamos los perfiles técnicos requeridos para el proyecto y los cruzamos con las habilidades actuales del equipo. Si detectamos una brecha de conocimiento, planificamos capacitaciones o mentorías dentro del plan.	Matriz de Capacidades / Plan de Capacitación: Registro de habilidades técnicas actuales del equipo y fechas programadas para entrenamientos.

Resultados_Evaluacion_entrevistas_CMMI_DEV_V20_ML2_GUION ENTREVISTA MANAGING

MANAGING	Planning and Managing Work	PLAN	PLAN 2.3	¿Cómo aseguran que el presupuesto financiero y el cronograma general de hitos se mantengan actualizados a lo largo de la ejecución?	No revisamos el avance en cada iteración, pero si se actualiza el calendario y las estimaciones financieras si existe modificación del alcance por parte del cliente.	Excel Financiero: Historial de Sprints planificados y reporte de costos mensuales o flujo de caja actualizado.
MANAGING	Planning and Managing Work	PLAN	PLAN 2.4	¿De qué manera planificaron la participación, involucración y frecuencia de reuniones de los diferentes interesados (stakeholders) del proyecto?	Identificamos a todos los stakeholders y planificamos su involucramiento mediante una matriz de interesados que define sus responsabilidades , y planificamos reuniones cada 15 días con carácter previo a las sesiones de sprint	Matriz de Interesados (Stakeholder Matrix): Listado de roles clave, sus responsabilidades asignadas y la frecuencia de reuniones asociadas. Almacenas en Drive
MANAGING	Planning and Managing Work	PLAN	PLAN 2.5	¿Cómo planificaron la entrega, despliegue o transición ordenada del software al equipo de operaciones o soporte técnico?	Desde la planificación inicial incluimos una estrategia de despliegue que detalla la entrega de manuales técnicos, la capacitación a la mesa de ayuda y los criterios de aceptación para el pase formal a producción.	Checklist de Despliegue: Documento con los criterios de aceptación de operaciones y actas de entrega firmadas.
MANAGING	Planning and Managing Work	PLAN	PLAN 2.6	¿Cómo demuestran que el plan de proyecto consolidado realmente es viable con los recursos y el tiempo que tienen disponibles?	Calculamos la capacidad real de horas de desarrollo de nuestro equipo frente al esfuerzo total estimado de las tareas del plan para garantizar que las fechas comprometidas sean realistas y factibles.	Excel en Drive: Matriz de Capacidad: Tabla de disponibilidad de horas por rol técnico que demuestra la conciliación y viabilidad de la asignación de recursos.
MANAGING	Planning and Managing Work	PLAN	PLAN 2.7	¿De que manera se especifica el plan del proyecto? ¿Como se la consistencia entre sus elementos? ¿Existe algún procedimiento de actualización?	Se crea un plan de proyecto, con estimación de esfuerzo y se asigna presupuesto y recursos, en el plan de proyecto se realiza una matriz de riesgo que se va actualizando en el curso del desarrollo, así mismo se crea un plan de pruebas y el plan de configuración. El plan se revisa de forma continua para evitar desviaciones indeseadas	En drive en un libro de excel plan de proyecto, matriz de riesgos,, plan de pruebas y plan de configuración
MANAGING	Planning and Managing Work	PLAN	PLAN 2.8	¿Cada cuanto tiempo se revisan los planes y se identifican las responsabilidades con los entregables?	Se crea una matriz de interesados y se vincula los distintos roles con los entregables. Se revisa en cada sprint, y se actualiza la matriz.	Matriz Rol-Entregable
MANAGING	Planning and Managing Work	MC	MC 1.1	De manera general, ¿cómo realiza la revisión o el seguimiento para verificar que las tareas completadas cumplan con lo solicitado originalmente?	Monitorizamos el avance de forma continua en las columnas de nuestro tablero de control. Al cierre de las actividades, revisamos que cada tarea finalizada cumpla con los criterios de aceptación técnicos establecidos.	Excel en drive: Panel general con filtros de tareas cerradas e historial de validación técnica de los ítems en estado 'Hecho'.

Resultados_Evaluacion_entrevistas_CMMI_DEV_V20_ML2_GUION ENTREVISTA MANAGING

MANAGING	Planning and Managing Work		MC 1.2	¿Cómo se identifican los problemas? ¿existen procedimientos estandarizados registrar los problemas?	Los problemas se identifican según su naturaleza de diferentes formas, si es de requerimiento, en las reuniones con los Stake holders, si son de código en los testing. Existe un procedimiento para la identificación única los problemas, con caracterización de los mismos estandarizada. así como un flujo definido para la adjudicación al técnico que corresponda.	Protocolo de caracterización de problemas. Matriz de registros de Issues
MANAGING	Planning and Managing Work	MC	MC 2.1	¿Cada cuánto tiempo comparan los resultados reales de esfuerzo, cronograma, recursos y presupuesto frente a lo que planificaron originalmente?	Lo hacemos quincenalmente en el cierre de iteración utilizando métricas de velocidad y gráficas de avance, y mensualmente mediante un reporte comparativo que contrasta las horas reales consumidas frente a la línea base planificada.	Excel Drive: Gráficas comparativos de 'Planificado vs. Real'.
MANAGING	Planning and Managing Work	MC	MC 2.2	¿Cómo realizan el monitoreo de los compromisos adquiridos y el nivel de involucración de las partes interesadas (clientes/usuarios)?	Llevamos un registro de los compromisos de los interesados (como la entrega de insumos, aprobaciones de diseño o asistencia a revisiones). Si hay un retraso de su parte, lo documentamos para evaluar su impacto en el cronograma.	Minutas de Reunión en Drive. Excel, registro de entregables, estado y retrasos.
MANAGING	Planning and Managing Work	MC	MC 2.3	¿Dónde monitorean los riesgos identificados del proyecto y cómo controlan la transición o entrega operativa para evitar problemas técnicos?	Mantenemos una matriz de riesgos activa donde evaluamos periódicamente la probabilidad e impacto de cada evento, definiendo planes de mitigación. Asimismo, monitoreamos los puntos críticos durante la fase de transición operativa.	Excel en Drive: Matriz de Riesgos (Risk Log) / Plan de Transición: Archivo actualizado con la lista de riesgos, dueños, planes de mitigación y alertas de pase a producción. En carpeta drive
MANAGING	Planning and Managing Work	MC	MC 2.4	Si detectan una desviación significativa (retraso crítico o sobrecosto), ¿dónde registran el problema y el análisis de la causa raíz?	Cuando un desvío supera nuestro límite de tolerancia, registramos el evento en una en una matriz de Issues, convocamos al equipo para analizar la causa raíz (usando los 5 Porqués) y estructuramos un plan de acción formal.	Excel Matriz de Issues/ Acta de Análisis de Causa Raíz: Documento que describe detalladamente la desviación encontrada y la causa técnica identificada.
MANAGING	Planning and Managing Work	MC	MC 2.5	¿Cómo demuestran que esas acciones correctivas realmente se ejecutaron, se les dio seguimiento y se gestionaron formalmente hasta su cierre?	Registramos las acciones correctivas como tareas formales en el Backlog con un responsable asignado. Realizamos el seguimiento continuo de su ejecución y no las damos por concluidas hasta verificar que mitigaron el desvío de forma efectiva.	Excel en Drive: identificamos estdos con etiquetas 'Acción Correctiva' o 'Mejora' transitados formalmente hasta el estado cerrado y vinculados al desvío original.

Formulario de Evidencias.

Se crea libro de Excel que contiene:

Una hoja por cada con una categoría que incluye las áreas de prácticas incluidas en dicha categoría y el conjunto de prácticas que conforman esa área de práctica para el cumplimiento del nivel de madurez 2 del modelo CMMI DEV 2.0 y que se recogen en la Que se encuentran recogidas en la Tabla 16. Listado de prácticas por PA's CMMI DEV V2.0 ML2 de este TFM

1- . Para cada una de las prácticas se incluyen 12 columnas

1. Capability Area
2. PA
3. Área de Práctica (PA)
4. Nivel
5. ID Práctica
6. Enunciado Oficial de la Práctica
7. Evidencia Directa
8. Evidencia Indirecta
9. ¿Cumple Proyecto A? (Sí/No/Falta)
10. ¿Cumple Proyecto B? (Sí/No/Falta)
11. Estado Cualitativo (FI / LI / PI / NI)
12. Brechas detectadas / Tareas pendientes

En el encabezado se informa del criterio para la determinación del nivel de cumplimiento de la práctica, el cumplimiento del área de práctica, del área de capacidad y de la categoría con el texto:

“CRITERIO OFICIAL DE SATISFACCIÓN (REGLA CMMI v2.0): El Área de Práctica (PA) estará calificada como SATISFECHA únicamente si el 100% de las Prácticas listadas arriba se encuentran totalmente cubiertas (FI o LI) dentro de las muestras

organizacionales. Un solo valor nulo (NI o PI) invalida y reprueba toda la PA de manera automática.”

Se muestra el resultado de la evaluación de evidencias para el área de Managing:

FORMULARIO DE EVALUACIÓN DE EVIDENCIAS CMMI DEV V2.0 ML2

CRITERIO OFICIAL DE SATISFACCIÓN (REGLA CMMI v2.0): El Área de Práctica (PA) estará calificada como SATISFECHA únicamente si el 100% de las Prácticas listadas arriba se encuentran totalmente cubiertas (FI o LI) dentro de las muestras organizacionales. Un solo valor nulo (NI o PI) invalida y reprueba toda la PA de manera automática.

Área de Capacidad	Práctica Area	Nivel	ID Práctica	Enunciado Oficial de la Práctica	Estado Cualitativo (FI /LI/ PI /N I)	Brechas detectadas / Tareas pendientes
Planning & Managing Work	EST	1	EST 1.1	Desarrollar estimaciones de alto nivel para el trabajo.	FI	Práctica satisfecha. Se conserva la línea base histórica del inicio del proyecto.
Planning & Managing Work	EST	2	EST 2.1	Estimar el alcance del trabajo basándose en una estructura de desglose de tareas (WBS).	FI	Práctica satisfecha. Todo el alcance está centralizado y jerarquizado sin requerimientos sueltos.
Planning & Managing Work	EST	2	EST 2.2	Estimar el esfuerzo, costo y cronograma del proyecto utilizando datos históricos o modelos.	FI	Práctica satisfecha. Se evidencia un método técnico de estimación relativa y participativa.
Planning & Managing Work	EST	2	EST 2.3	Registrar y mantener las bases y suposiciones de las estimaciones.	FI	Práctica satisfecha. Existe una lógica matemática que traduce los Story Points a costos y plazos reales.
	EST				SATISFIED	
Planning & Managing Work	PLAN	1	PLAN 1.1	Desarrollar una lista de tareas para realizar el trabajo.	FI	Práctica satisfecha. El arranque cuenta con una distribución formal de actividades iniciales.
Planning & Managing Work	PLAN	2	PLAN 1.2	Asignar personas a las tareas.	FI	Práctica satisfecha. La asignación se basa en la capacidad del personal técnico.
Planning & Managing Work	PLAN	2	PLAN 2.1	Desarrollar y mantener el enfoque y plan para cumplir con el trabajo.	FI	Práctica satisfecha. El ciclo de vida responde a las características de este proyecto y está formalizado.
Planning & Managing Work	PLAN	2	PLAN 2.2	Desarrollar y mantener el presupuesto y el cronograma del proyecto.	FI	Práctica satisfecha. Existe un mapeo de brechas de conocimiento y acciones formativas en marcha.
Planning & Managing Work	PLAN	2	PLAN 2.3	Planificar la asignación de recursos y el conocimiento necesario para el proyecto.	LI	Brecha menor: Las actualizaciones financieras y de calendario no se hacen en cada iteración rutinaria, solo ante cambios de alcance del cliente.

FORMULARIO DE EVALUACIÓN DE EVIDENCIAS CMMI DEV V2.0 ML2

Planning & Managing Work	PLAN	2	PLAN 2.4	Identificar e involucrar a las partes interesadas relevantes en la planificación.	FI	Práctica satisfecha. La comunicación y gobernanza con el cliente están totalmente planificadas.
Planning & Managing Work	PLAN	2	PLAN 2.5	Planificar la transición a operaciones y soporte.	FI	Práctica satisfecha. El paso a producción cuenta con una estrategia de transición documentada.
Planning & Managing Work	PLAN	2	PLAN 2.6	Obtener el compromiso de los responsables de ejecutar el plan del proyecto.	FI	Práctica satisfecha. Se comprueba matemáticamente que las fechas y entregas prometidas son factibles.
Planning & Managing Work	PLAN	2	PLAN 2.7	Desarrollar el plan del proyecto, asegurar la consistencia entre sus elementos y mantenerlo actualizado.	FI	Práctica satisfecha. El plan actúa como un ecosistema vivo e integrado, no como documentos aislados.
Planning & Managing Work	PLAN	2	PLAN 2.8	Revisar los planes y obtener compromisos de las partes interesadas afectadas.	FI	Práctica satisfecha. Existe una correlación formal entre quién responde por cada componente técnico.
	PLAN				SATISFIED	
Planning & Managing Work	MC	1	MC 1.1	Revisar las tareas completadas y los resultados del trabajo.	FI	Práctica satisfecha. El seguimiento operativo en el tablero asegura que lo terminado cumple los requisitos.
Planning & Managing Work	MC	2	MC 1.2	Identificar y resolver problemas.	FI	Práctica satisfecha. Existe una vía uniforme y conocida para clasificar y resolver incidencias del código.
Planning & Managing Work	MC	2	MC 2.1	Monitorear los resultados reales del proyecto contra el plan (cronograma, costos, alcance).	FI	Práctica satisfecha. Las desviaciones en tiempo y costo se miden periódicamente a dos niveles.
Planning & Managing Work	MC	2	MC 2.2	Monitorear el involucramiento de las partes interesadas.	FI	Práctica satisfecha. Se mide el impacto real que tienen los retrasos del cliente en el cronograma.
Planning & Managing Work	MC	2	MC 2.3	Monitorear la transición a operaciones y soporte.	FI	Práctica satisfecha. Los riesgos del proyecto y los puntos críticos del despliegue se evalúan continuamente.
Planning & Managing Work	MC	2	MC 2.4	Identificar, analizar y documentar las desviaciones significativas.	FI	Práctica satisfecha. Ante desvíos graves, se frena la operación para identificar y documentar el origen del fallo.

FORMULARIO DE EVALUACIÓN DE EVIDENCIAS CMMI DEV V2.0 ML2

Planning & Managing Work	MC	2	MC 2.5	Tomar acciones correctivas cuando el desempeño varíe significativamente y seguirlas hasta su cierre.	FI	Práctica satisfecha. Las acciones correctivas no se archivan, se vuelven tareas obligatorias del backlog.
	MC				SATISFIED	
Planning & Managing Work					SATISFIED	
Managing					SATISFIED	

Anexo II Matriz de relación prácticas CMMI V2.0

Adaptado y traducido de (CMMI Institute, 2019)

Práctica	Descripción
CAR 1.1	Identificar y abordar las causas de los resultados seleccionados.
CAR 2.1	Seleccionar los resultados para análisis.
CAR 2.2	Analizar y abordar las causas de los resultados.
CAR 3.1	Determinar las causas raíz de los resultados seleccionados siguiendo un proceso organizacional.
CAR 3.2	Proponer acciones para abordar las causas raíz identificadas.
CAR 3.3	Implementar las propuestas de acción seleccionadas.
CAR 3.4	Registrar datos de análisis y resolución de causas raíz.
CAR 3.5	Presentar propuestas de mejora para cambios demostrados como eficaces.
CAR 4.1	Realizar análisis de causa raíz de los resultados seleccionados, aplicando técnicas estadísticas y técnicas cuantitativas de otro tipo.
CAR 4.2	Evaluar el efecto de acciones implementadas en el rendimiento del proceso, aplicando técnicas estadísticas y técnicas cuantitativas de otro tipo
CAR 5.1	Utilizar técnicas estadísticas y técnicas cuantitativas de otro tipo para evaluar otras soluciones y procesos para determinar si debe aplicarse la resolución en una escala más amplia.
CM 1.1	Realizar el control de versiones.
CM 2.1	Identificar los artículos a ser incluidos en la Gestión de la configuración.
CM 2.2	Desarrollar, actualizar y usar un sistema de configuración y de Gestión del cambio.
CM 2.3	Desarrollar o liberar líneas base para uso interno o para su entrega al cliente.
CM 2.4	Gestionar los cambios en los elementos de la gestión de la configuración.
CM 2.5	Desarrollar, mantener actualizados y utilizar registros que describen los elementos de la gestión de la configuración.
CM 2.6	Realizar auditorías de configuración para mantener la integridad de las líneas base de configuración, cambios y contenido del sistema de gestión de la configuración.
CONT 1.1	Desarrollar enfoques de contingencia para gestionar interrupciones relevantes en las operaciones.
CONT 2.1	Identificar y priorizar las funciones esenciales para la continuidad.

CONT 2.2	Identificar y priorizar de los recursos esenciales para la continuidad.
CONT 2.3	Desarrollar, mantener actualizados y seguir los planes de continuidad para reanudar la realización de las funciones esenciales.
CONT 3.1	Desarrollar y mantener actualizados los materiales para la capacitación en continuidad.
CONT 3.2	Impartir y evaluar la de capacitación en continuidad de acuerdo con el plan.
CONT 3.3	Preparar, llevar a cabo y analizar los resultados de la verificación y la validación del plan de continuidad.
DAR 1.1	Definir y registrar alternativas.
DAR 1.2	Crear y registrar la decisión.
DAR 2.1	Desarrollar, mantener actualizado y utilizar reglas para determinar cuándo seguir un proceso registrado para la toma de decisiones basadas en criterios.
DAR 2.2	Desarrollar criterios para evaluar alternativas.
DAR 2.3	Identificar soluciones alternativas.
DAR 2.4	Seleccionar métodos de evaluación.
DAR 2.5	Evaluar y seleccionar las soluciones mediante criterios y métodos.
DAR 3.1	Desarrollar, mantener actualizados y usar una descripción de una autoridad de decisiones, basada en roles.
EST 1.1	Desarrollar estimaciones de alto nivel para realizar el trabajo.
EST 2.1	Desarrollar, mantener actualizada y utilizar el alcance de lo que estima.
EST 2.2	Desarrollar y mantener actualizadas las estimaciones para el tamaño de la solución.
EST 2.3	En base a estimaciones de tamaño, desarrollar y registrar el esfuerzo, la duración y las estimaciones de costos y su razonamiento para la solución.
EST 3.1	Desarrollar y mantener actualizado un método de estimación registrada.
EST 3.2	Utilizar los activos de repositorio y el proceso de medición organizativa para la estimación de trabajo.
GOV 1.1	La alta gerencia identifica lo que es importante para realizar el trabajo y define el enfoque necesario para lograr los objetivos de la organización.
GOV 2.1	La alta gerencia define, mantiene actualizada y comunica las directivas organizacionales para la implementación de procesos y mejora basadas en objetivos y necesidades de la organización.

GOV 2.2	La alta gerencia asegura que se proporcionen recursos y capacitación para desarrollar, apoyar, realizar, mejorar y evaluar el cumplimiento de los procesos esperados.
GOV 2.3	La alta gerencia identifica sus necesidades de información y utiliza la información recopilada para proporcionar gobernanza y supervisar eficazmente la implementación y mejora de procesos.
GOV 2.4	La alta gerencia asigna responsabilidad a las personas por el cumplimiento de las directivas de la organización y por lograr la implementación de procesos y objetivos de mejora.
GOV 3.1	La alta gerencia asegura que las medidas que apoyan los objetivos de la organización se recopilen, analicen y utilicen.
GOV 3.2	La alta gerencia asegura que las competencias y procesos estén alineados con los objetivos de la organización.
GOV 4.1	La alta gerencia garantiza que las decisiones seleccionadas estén impulsadas por análisis estadísticos y cuantitativos relacionados con el desempeño y logro de los objetivos de calidad y rendimiento del proceso
II 1.1	Realizar procesos que aborden la intención de las prácticas del nivel 1.
II 2.1	Proporcionar recursos suficientes, financiación y capacitación para el desarrollo y la ejecución de los procesos.
II 2.2	Desarrollar y mantener los procesos actualizados, y verificar que se cumplan.
II 3.1	Utilizar procesos organizacionales y activos de procesos para planificar, administrar y realizar el trabajo.
II 3.2	Evaluar el cumplimiento y la eficacia de los procesos organizacionales.
II 3.3	Contribuir con información relacionada al proceso o con activos de procesos a la organización.
IRP 1.1	Registrar y resolver incidencias.
IRP 2.1	Desarrollar, mantener actualizado y seguir un enfoque para la resolución y prevención de incidentes.
IRP 2.2	Supervisar y resolver cada incidente hasta su conclusión.
IRP 2.3	Comunicar el estado de los incidentes
IRP 3.1	Desarrollar, mantener actualizado y utilizar un sistema de gestión de incidentes para el
IRP 3.2	Analizar datos de incidentes seleccionados y la resolución para la prevención de futuros incidentes.

MC 1.1	Registrar los términos de las tareas.
MC 1.2	Identificar y resolver problemas.
MC 2.1	Realizar un seguimiento de los resultados actuales y compararlos con las estimaciones en tamaño, esfuerzo, programación, recursos, conocimientos y habilidades y presupuesto.
MC 2.2	Realizar un seguimiento a la implicación de los grupos involucrados y compromisos identificados.
MC 2.3	Supervisar la transición de las operaciones y soporte técnico.
MC 2.4	Tomar medidas correctivas cuando los resultados reales difieren significativamente de los resultados planeados y gestionarlos hasta el cierre.
MC 3.1	Gestionar el proyecto con el plan de proyecto y el proceso del proyecto.
MC 3.2	Gestionar dependencias y actividades fundamentales.
MC 3.3	Supervisar el entorno de trabajo para identificar problemas.
MC 3.4	Gestionar y resolver problemas con los grupos involucrados afectados
MPM 1.1	Recopilar medidas y registrar el rendimiento. MPM 1.2
MPM 1.2	Identificar y resolver problemas de rendimiento.
MPM 2.1	Derivar y registrar los objetivos de medición y rendimiento basados en necesidades y objetivos seleccionados del negocio y mantenerlos actualizados.
MPM 2.2	Desarrollar, mantener actualizada y utilizar definiciones operacionales para las medidas.
MPM 2.3	Obtener datos de medida específicos según las definiciones operacionales.
MPM 2.4	Analizar datos de rendimiento y medición según las definiciones operacionales.
MPM 2.5	Almacenar datos de medición, especificaciones de medición y resultados de análisis según las definiciones operacionales.
MPM 2.6	Tomar acciones para resolver problemas identificados con el cumplimiento de los objetivos de medición y rendimiento.
MPM 3.1	Desarrollar, mantener actualizada y utilizar objetivos de medición y rendimiento organizacionales con trazabilidad a los objetivos del negocio.
MPM 3.2	Seguir los procesos y estándares organizacionales para desarrollar y utilizar definiciones operacionales para las medidas y mantenerlos actualizados.
MPM 3.3	Desarrollar, mantener actualizados y seguir un proceso de calidad de datos.
MPM 3.4	Desarrollar, mantener actualizada y utilizar el repositorio de mediciones de la organización.

MPM 3.5	Analizar el rendimiento organizacional utilizando la medición y datos del rendimiento para determinar las necesidades de mejora del rendimiento.
MPM 3.6	Comunicar periódicamente los resultados de rendimiento a la organización.
MPM 4.1	Utilizar técnicas estadísticas y técnicas cuantitativas de otro tipo para desarrollar, mantener actualizados y comunicar los objetivos de calidad y rendimiento del proceso con trazabilidad a los objetivos del negocio.
MPM 4.2	Seleccionar medidas y técnicas analíticas para gestionar/administrar cuantitativamente el rendimiento para alcanzar los objetivos de calidad y rendimiento del proceso.
MPM 4.3	Utilizar técnicas estadísticas y técnicas cuantitativas de otro tipo para desarrollar y analizar las líneas base de rendimiento del proceso(s) y mantenerlas actualizadas.
MPM 4.4	Utilizar técnicas estadísticas y técnicas cuantitativas de otro tipo para desarrollar y analizar las líneas base de rendimiento del proceso(s) y mantenerlas actualizadas.
MPM 4.5	Utilizar técnicas estadísticas y técnicas cuantitativas de otro tipo para determinar o predecir el logro de los objetivos de calidad y rendimiento del proceso.
MPM 5.1	Utilizar técnicas estadísticas y técnicas cuantitativas de otro tipo para garantizar que los objetivos del negocio se alinean con la estrategia y rendimiento del negocio.
MPM 5.2	Analizar datos del rendimiento mediante técnicas estadísticas y técnicas cuantitativas de otro tipo para determinar la capacidad de la organización de satisfacer los objetivos del negocio seleccionados e identificar posibles áreas para la mejora del rendimiento.
MPM 5.3	Seleccionar e implementar las propuestas de mejora, basadas en el análisis estadístico y cuantitativo del efecto esperado de las mejoras propuestas en el cumplimiento de los objetivos de negocio, calidad y rendimiento del proceso
OT 1.1	Capacitar al personal.
OT 2.1	Identificar las necesidades de capacitación.
OT 2.2	Capacitar al personal y mantener registros.
OT 3.1	Desarrollar y mantener actualizadas las necesidades de capacitación estratégicas y a corto plazo de la organización.
OT 3.2	Coordinar las necesidades de capacitación y de entrega entre los proyectos y la organización.

OT 3.3	Desarrollar, mantener actualizados y seguir los planes de capacitación estratégica y a corto plazo de la organización.
OT 3.4	Desarrollar, mantener actualizada y utilizar una capacidad de capacitación para abordar las necesidades de aprendizaje de la organización.
OT 3.5	Evaluar la efectividad del programa de capacitación de la organización.
OT 3.6	Registrar, mantener actualizado y utilizar el conjunto de registros de capacitación organizacional
PAD 1.1	Desarrollar activos del proceso para realizar el trabajo.
PAD 2.1	Determinar qué activos del proceso se necesitarán para realizar el trabajo.
PAD 2.2	Desarrollar, comprar o reutilizar los activos del proceso. PAD 2.3
PAD 2.3	Hacer disponibles los procesos y activos.
PAD 3.1	Desarrollar, mantener actualizada y seguir una estrategia para la creación y actualización de activos del proceso.
PAD 3.2	Desarrollar, registrar y mantener actualizada una arquitectura de proceso que describa la estructura de los procesos y los activos de los procesos de la organización.
PAD 3.3	Desarrollar, mantener actualizados y realizar procesos y activos disponibles para su uso.
PAD 3.4	Desarrollar, mantener actualizados y utilizar criterios y guías de adaptación para el conjunto de procesos estándar y activos.
PAD 3.5	Desarrollar, mantener actualizada y hacer disponible la biblioteca de activos de procesos de la organización para su uso.
PAD 3.6	Desarrollar, mantener actualizada y hacer disponible los estándares de entorno de trabajo para su uso.
PAD 3.7	Desarrollar, mantener actualizados y hacer disponibles los estándares de medición y análisis de la organización para su uso.
PCM 1.1	Desarrollar una estructura de soporte para proporcionar orientación sobre los procesos, identificar y solucionar problemas en los procesos y mejorar continuamente los procesos.
PCM 1.2	Evaluar la implementación actual de los procesos e identificar fortalezas y debilidades.
PCM 1.3	Abordar las oportunidades de mejora o los problemas en los procesos.
PCM 2.1	Identificar mejoras en los procesos y los activos de los procesos.

PCM 2.2	Desarrollar, mantener actualizados y seguir planes para implementar mejoras seleccionadas a los procesos.
PCM 3.1	Desarrollar, mantener actualizadas y usar mejoras de proceso, trazables a objetivos de negocio.
PCM 3.2	Identificar los procesos que más contribuyen en cumplir los objetivos de negocio.
PCM 3.3	Explorar y evaluar posibles nuevos procesos, técnicas, métodos y herramientas para identificar oportunidades de mejora.
PCM 3.4	Proporcionar soporte para implementar, desplegar y sostener las mejoras a los procesos.
PCM 3.5	Desplegar procesos estándar y activos de procesos de la organización
PCM 3.6	Evaluar la efectividad de las mejoras desplegadas en el logro de los objetivos de mejora de procesos.
PCM 4.1	Usar técnicas estadísticas y técnicas cuantitativas de otro tipo para validar mejoras seleccionadas de desempeño contra expectativas de mejora propuestas, objetivos de negocio u objetivos de calidad y rendimiento del proceso.
PI 1.1	Ensamblar soluciones y entregarlas al cliente.
PI 2.1	Desarrollar una estrategia de integración, mantenerla actualizada y seguirla.
PI 2.2	Desarrollar el ambiente de integración, mantenerlo actualizado y utilizarlo.
PI 2.3	Desarrollar procedimientos y criterios para la integración de soluciones y componentes, mantenerlos actualizados y seguirlos.
PI 2.4	Confirmar, antes de la integración, que cada componente haya sido identificado correctamente y funcione de acuerdo con sus requisitos y diseño.
PI 2.5	Evaluar los componentes integrados para garantizar el cumplimiento de los requisitos y el diseño de la solución.
PI 2.6	Integrar soluciones y componentes según la estrategia de integración.
PI 3.1	Revisar y mantener actualizadas las descripciones de interfaz o conexión, en cuanto a cobertura, completitud y consistencia a lo largo de la vida de la solución.
PI 3.2	Confirmar, antes de la integración, que las interfaces de los componentes o conexiones cumplen con las descripciones de interfaz o conexión.
PI 3.3	Evaluar los componentes integrados para garantizar el cumplimiento de los requisitos y el diseño de la solución.

PLAN 1.1	Desarrollar una lista de tareas.
PLAN 1.2	Asignar personas a las tareas.
PLAN 2.2	Desarrollar y mantener actualizado el enfoque para llevar a cabo el trabajo.
PLAN 2.2	Planificar los conocimientos y capacidades necesarios para realizar el trabajo.
PLAN 2.3	Basándose en estimaciones registradas, desarrollar y mantener el presupuesto y el calendario actualizado.
PLAN 2.4	Planificar la implicación de las partes interesadas identificadas.
PLAN 2.5	Planificar la transición a las operaciones y soporte técnico.
PLAN 2.6	Asegurarse de que los planes son factibles al conciliar los recursos disponibles y estimados.
PLAN 2.7	Desarrollar el plan de proyecto, garantizar la coherencia entre sus elementos y mantenerlo actualizado.
PLAN 2.8	Revisar los planes y obtener compromisos de los grupos involucrados afectados.
PLAN 3.1	Utilizar el conjunto de procesos estándar de la organización y guías de adaptación para desarrollar, mantener actualizado y seguir el proceso del proyecto.
PLAN 3.2	Desarrollar un plan y mantenerlo actualizado, utilizando el proceso del proyecto, los activos de procesos de la organización y el repositorio de medición.
PLAN 3.3	Identificar y negociar las dependencias críticas.
PLAN 3.4	Planificar el entorno del proyecto y mantenerlo actualizado en base a los estándares de la organización.
PLAN 4.1	Utilizar técnicas estadísticas y técnicas cuantitativas de otro tipo para desarrollar y mantener los procesos
PQA 1.1	Identificar y solventar problemas con los procesos y productos de trabajo.
PQA 2.1	Desarrollar, mantener actualizado y seguir un planteamiento de aseguramiento de la calidad y un plan acorde basado en los datos históricos de calidad.
PQA 2.2	Durante el proyecto, evaluar objetivamente procesos ejecutados seleccionados y sus productos de trabajo contra el proceso documentado y los estándares aplicables
PQA 2.3	Comunicar los problemas de calidad y no-conformidades y asegurar su resolución
PQA 2.4	Registrar y utilizar los resultados de las actividades de aseguramiento de la calidad.
PQA 3.1	Identificar y registrar las oportunidades de mejora durante las actividades de aseguramiento de la calidad

PR 1.1	Realizar revisiones de productos de trabajo y registrar los problemas
PR 2.1	Desarrollar y mantener actualizados los procedimientos y materiales de apoyo que se utiliza para preparar y realizar revisiones por pares.
PR 2.2	Seleccionar los productos de trabajo para ser revisados por pares.
PR 2.3	Preparar y realizar revisiones por pares en productos de trabajo seleccionados aplicando procedimientos establecidos.
PR 2.4	Resolver los problemas identificados en las revisiones por pares.
PR 3.1	Analizar resultados y datos de las revisiones por pares.
RDM 1.1	Registrar requerimientos.
RDM 2.1	Recabar las necesidades, expectativas, restricciones e interfaces o conexiones de todas las partes interesadas
RDM 2.2	Transformar las necesidades, expectativas, restricciones y las interfaces o conexiones de las partes interesadas en requerimientos priorizados del cliente.
RDM 2.3	Desarrollar un entendimiento con los proveedores de requerimiento sobre el significado de los mismos
RDM 2.4	Obtener el compromiso de los participantes del proyecto, de que pueden implementar los requerimientos.
RDM 2.5	Desarrollar, registrar y mantener trazabilidad bidireccional entre los requerimientos y las actividades o productos de trabajo.
RDM 2.6	Asegurar que los planes y actividades o productos de trabajo continúen siendo congruentes con los requerimientos.
RDM 3.1	Desarrollar y mantener actualizados los requerimientos correspondientes a la solución y sus componentes.
RDM 3.2	Desarrollar conceptos y escenarios operativos.
RDM 3.3	Desarrollar conceptos y escenarios operacionales.
RDM 3.4	Identificar, desarrollar y mantener actualizados los requerimientos de interfaz o conexión.
RDM 3.5	Asegurar que los requerimientos sean necesarios y suficientes.
RDM 3.6	Equilibrar las necesidades y las restricciones de los grupos involucrados.
RDM 3.7	Validar los requerimientos para garantizar que la solución resultante va a funcionar como se esperaba en el ambiente de destino.

RSK 1.1	Identificar y registrar los riesgos u oportunidades y mantenerlos actualizados.
RSK 2.1	Analizar los riesgos u oportunidades identificados.
RSK 2.2	Monitorear los riesgos u oportunidades identificados y comunicar el estado a los grupos involucrados afectados.
RSK 3.1	Identificar y utilizar las categorías de riesgo u oportunidad.
RSK 3.2	Definir y utilizar los parámetros para el manejo y análisis de riesgos u oportunidades.
RSK 3.3	Desarrollar y mantener actualizada una estrategia de gestión de riesgos u oportunidades.
RSK 3.4	Desarrollar y mantener actualizados los planes de gestión de riesgos u oportunidades.
RSK 3.5	Gestionar los riesgos u oportunidades mediante la implementación de actividades planificadas de administración de riesgos u oportunidades
SAM 1.1	Desarrollar y registrar el acuerdo con el proveedor.
SAM 1.2	Aceptar o rechazar los entregables del proveedor.
SAM 1.3	Procesar facturas de proveedores.
SAM 2.1	Supervisar al proveedor como se especifica en el acuerdo con el proveedor y mantener actualizado el acuerdo.
SAM 2.2	Realizar las actividades que se especifiquen en el acuerdo con el proveedor.
SAM 2.3	Verificar que se cumpla el acuerdo con el proveedor antes de aceptar el entregable adquirido del proveedor.
SAM 2.4	Administrar las facturas que envíe el proveedor según los acuerdos con el proveedor.
SAM 3.1	Seleccionar los entregables técnicos del proveedor para su análisis y llevar a cabo revisiones técnicas.
SAM 3.2	Seleccionar y supervisar los procesos y entregables del proveedor en base a los criterios del acuerdo con el proveedor.
SAM 4.1	Seleccionar medidas y aplicar técnicas analíticas para gestionar cuantitativamente el rendimiento del proveedor para alcanzar los objetivos de calidad y rendimiento del proceso.
SDM 1.1	Utilizar el sistema de servicio para ofrecer servicios.
SDM 2.1	Desarrollar, registrar, mantener actualizados y seguir los acuerdos de servicio.
SDM 2.2	Recibir y procesar las solicitudes de servicio conforme a los acuerdos de servicio.
SDM 2.3	Ofrecer servicios conforme a los acuerdos de servicio.

SDM 2.4	Analizar los acuerdos de servicio existentes y los datos de servicio para preparar acuerdos nuevos o actualizados.
SDM 2.5	Desarrollar, registrar, mantener actualizado y seguir el enfoque para operar y cambiar el sistema de servicio.
SDM 2.6	Confirmar la disponibilidad del sistema de servicio para soportar la entrega de servicios.
SDM 3.1	Desarrollar, registrar, mantener actualizados y usar los sistemas y acuerdos de servicio estándar organizacionales.
SSS 1.1	Determinar el tipo de adquisición.
SSS 1.2	Identificar proveedores potenciales y distribuir solicitudes de propuestas.
SSS 1.3	Evaluar propuestas y seleccionar proveedores.
SSS 2.1	Desarrollar un paquete de solicitud y mantenerlo actualizado.
SSS 2.2	Identificar proveedores potenciales calificados y distribuir el paquete de solicitud para su respuesta.
SSS 2.3	Evaluar las soluciones propuestas de acuerdo con los criterios de evaluación registrados y selección de proveedores.
SSS 3.1	Desarrollar, mantener actualizados y seguir los enfoques de negociación para solicitar, evaluar y seleccionar proveedores.
STSM 1.1	Desarrollar una lista de los servicios existentes.
STSM 2.1	Desarrollar, mantener actualizadas y utilizar descripciones de servicios actuales.
STSM 2.2	Recopilar, registrar y analizar datos acerca de las necesidades estratégicas y capacidades para la prestación de servicios.
STSM 2.3	Desarrollar, mantener actualizado y seguir un enfoque de prestación de servicios nuevos o modificados derivados de las necesidades y capacidades estratégicas.
STSM 3.1	Desarrollar, mantener actualizados y utilizar el conjunto de servicios estándar organizacional y
TS 1.1	Construir una solución para cumplir con requerimientos.
TS 2.1	Diseñar y construir una solución que cumpla con los requerimientos.
TS 2.2	Evaluar el diseño y solventar los problemas identificados.
TS 2.3	Proporcionar orientación sobre el uso de la solución
TS 3.1	Desarrollar criterios para las decisiones de diseño.
TS 3.2	Desarrollar soluciones alternativas para los componentes seleccionados.
TS 3.3	Realizar un análisis de creación, compra o reutilización

TS 3.4	Seleccionar soluciones basadas en los criterios de diseño.
TS 3.5	Desarrollar, mantener actualizada y utilizar la información necesaria para implementar el diseño.
TS 3.6	Diseñar interfaces o conexiones de la solución con los criterios establecidos.
VV 1.1	Realizar las actividades de verificación para asegurar que los requisitos son implementados y registrados y comunican los resultados.
VV 1.2	Realizar la validación para asegurar que la solución funcionará como se esperaba en el entorno de operación y registrar y comunicar los resultados.
VV 2.1	Seleccionar los componentes y métodos para la verificación y la validación.
VV 2.2	Desarrollar, actualizar y utilizar el entorno necesario para realizar la verificación y la validación.
VV 2.3	Desarrollar, actualizar y seguir los procedimientos para la verificación y la validación
VV 3.1	Desarrollar, actualizar y utilizar los criterios para la verificación y la validación.
VV 3.2	Analizar y comunicar los resultados de la verificación y la validación

Anexo III: Matriz Prácticas CMMI DEV V2.0.

Adaptado de (CMMI Institute, 2018)

Practice Area	Level	Practice	Practice (description)
CAR	1	CAR 1.1	Identify and address causes of selected outcomes
CAR	2	CAR 2.1	Select outcomes for analysis
CAR	2	CAR 2.2	Analyze and address causes of outcomes
CAR	3	CAR 3.1	Determine root causes of selected outcomes by following an organizational process
CAR	3	CAR 3.2	Propose actions to address identified root causes
CAR	3	CAR 3.3	Implement selected action proposals
CAR	3	CAR 3.4	Record root cause analysis and resolution data
CAR	3	CAR 3.5	Submit improvement proposals for changes proven to be effective
CAR	4	CAR 4.1	Perform root cause analysis of selected outcomes using statistical and other quantitative techniques
CAR	4	CAR 4.2	Evaluate the effect of implemented actions on process performance using statistical and other quantitative techniques
CAR	5	CAR 5.1	Use statistical and other quantitative techniques to evaluate other solutions and processes to determine if the resolution should be applied on a broader scale
CM	1	CM 1.1	Perform version control.
CM	2	CM 2.1	Identify items to be placed under configuration management.
CM	2	CM 2.2	Develop, keep updated, and use a configuration and change management system.
CM	2	CM 2.3	Develop or release baselines for internal use or for delivery to the customer. CM 2.4
CM	2	CM 2.5	Develop, keep updated, and use records describing items under configuration management.
CM	2	CM 2.6	Perform configuration audits to maintain the integrity of configuration baselines, changes, and content of the configuration management system.
DAR	1	DAR 1.1	Define and record the alternatives
DAR	1	DAR 1.2	Make and record the decision
DAR	2	DAR 2.1	Develop, keep updated, and use rules to determine when to follow a recorded process for criteria-based decisions
DAR	2	DAR 2.2	Develop criteria for evaluating alternatives
DAR	2	DAR 2.3	Identify alternative solutions
DAR	2	DAR 2.4	Select evaluation methods
DAR	2	DAR 2.5	Evaluate and select solutions using criteria and methods
DAR	3	DAR 3.1	Develop, keep updated, and use a description of role-based decision authority
EST	1	EST 1.1	Develop high-level estimates to perform the work

EST	2	EST 2.1	Develop, keep updated, and use the scope of what is being estimated
EST	2	EST 2.2	Develop and keep updated estimates for the size of the solution
EST	2	EST 2.3	Based on size estimates, develop and record effort, duration, and cost estimates and their rationale for the solution
EST	3	EST 3.1	Develop and keep updated a recorded estimation method
EST	3	EST 3.2	Use the organizational measurement repository and process assets for estimating work
GOV	1	GOV 1.1	Senior management identifies what is important for doing the work and defines the approach needed to accomplish the objectives of the organization
GOV	2	GOV 2.1	Senior management defines, keeps updated, and communicates organizational directives for process implementation and improvement based on organization needs and objectives
GOV	2	GOV 2.2	Senior management ensures resources are provided for developing, supporting, performing, improving, and evaluating adherence to expected processes
GOV	2	GOV 2.3	Senior management identifies their information needs and uses the collected information to provide governance and oversight of effective process implementation and improvement
GOV	2	GOV 2.4	Senior management holds people accountable for adhering to organization directives and achieving process implementation and improvement objectives
GOV	3	GOV 3.1	Senior management ensures that measures supporting objectives throughout the organization are collected, analyzed, and used
GOV	3	GOV 3.2	Senior management ensures that competencies and processes are aligned with the objectives of the organization
GOV	4	GOV 4.1	Senior management ensures that selected decisions are driven by statistical and quantitative analysis related to performance and achievement of quality and process performance objectives
II	1	II 1.1	Perform processes that address the intent of the Level 1 practices
II	2	II 2.1	Provide sufficient resources, funding, and training for developing processes
II	2	II 2.2	Develop and keep processes updated, and verify they are being followed
II	3	II 3.1	Use organizational processes and process assets to plan, manage, and perform the work
II	3	II 3.2	Evaluate the adherence to and effectiveness of the organizational processes
II	3	II 3.3	Contribute process-related information or process assets to the organization

MPM	1	MPM 1.1	Collect measures and record performance
MPM	1	MPM 1.2	Identify and address performance issues
MPM	2	MPM 2.1	Derive and record measurement and performance objectives from selected business needs and objectives and keep them updated
MPM	2	MPM 2.2	Develop, keep updated, and use operational definitions for measures
MPM	2	MPM 2.3	Obtain specified measurement data according to the operational definitions
MPM	2	MPM 2.4	Analyze performance and measurement data according to the operational definitions
MPM	2	MPM 2.5	Store measurement data, measurement specifications, and analysis results according to the operational definitions
MPM	2	MPM 2.6	Take actions to address identified issues with meeting measurement and performance objectives
MPM	3	MPM 3.1	Develop, keep updated, and use organizational measurement and performance objectives traceable to business objectives
MPM	3	MPM 3.2	Follow organizational processes and standards to develop and use operational definitions for measures and keep them updated
MPM	3	MPM 3.3	Develop, keep updated, and follow a data quality process
MPM	3	MPM 3.4	Develop, keep updated, and use the organization's measurement repository
MPM	3	MPM 3.5	Analyze organizational performance using measurement and performance data to determine performance improvement needs
MPM	3	MPM 3.6	Periodically communicate performance results to the organization
MPM	4	MPM 4.1	Use statistical and other quantitative techniques to develop, keep updated, and communicate quality and process performance objectives that are traceable to business objectives
MPM	4	MPM 4.2	Select measures and analytic techniques to quantitatively manage performance to achieve quality and process performance objectives
MPM	4	MPM 4.3	Use statistical and other quantitative techniques to develop and analyze process performance baselines and keep them updated
MPM	4	MPM 4.4	Use statistical and other quantitative techniques to develop and analyze process performance models and keep them updated
MPM	4	MPM 4.5	Use statistical and other quantitative techniques to determine or predict achievement of quality and process performance objectives
MPM	5	MPM 5.1	Use statistical and other quantitative techniques to ensure that business objectives are aligned with business strategy and performance

MPM	5	MPM 5.2	Analyze performance data using statistical and other quantitative techniques to determine the organization's ability to satisfy selected business objectives and identify potential areas for performance improvement
MPM	5	MPM 5.3	Select and implement improvement proposals, based on the statistical and quantitative analysis of the expected effect of proposed improvements on meeting business, quality, and process performance objectives
MC	1	MC 1.1	Record task completions
MC	1	MC 1.2	Identify and resolve issues
MC	2	MC 2.1	Track actual results against estimates for size, effort, schedule, resources, knowledge and skills, and budget
MC	2	MC 2.2	Track the involvement of identified stakeholders and commitments
MC	2	MC 2.3	Monitor the transition to operations and support
MC	2	MC 2.4	Take corrective actions when actual results differ significantly from planned results and manage to closure
MC	3	MC 3.1	Manage the project using the project plan and the project process
MC	3	MC 3.2	Manage critical dependencies and activities
MC	3	MC 3.3	Monitor the work environment to identify issues
MC	3	MC 3.4	Manage and resolve issues with affected stakeholders
OT	1	OT 1.1	Train people
OT	2	OT 2.1	Identify training needs
OT	2	OT 2.2	Train personnel and keep records
OT	3	OT 3.1	Develop and keep updated the organization's strategic and short-term training needs
OT	3	OT 3.2	Coordinate training needs and delivery between the projects and the organization
OT	3	OT 3.3	Develop, keep updated, and follow organizational strategic and short-term training plans
OT	3	OT 3.4	Develop, keep updated, and use a training capability to address organizational training needs
OT	3	OT 3.5	Assess the effectiveness of the organization's training program
OT	3	OT 3.6	Record, keep updated, and use the set of organizational training records
PR	1	PR 1.1	Perform reviews of work products and record issues
PR	2	PR 2.1	Develop and keep updated procedures and supporting materials used to prepare for and perform peer reviews
PR	2	PR 2.2	Select work products to be peer reviewed
PR	2	PR 2.3	Prepare and perform peer reviews on selected work products using established procedures
PR	2	PR 2.4	Resolve issues identified in peer reviews

PR	3	PR 3.1	Analyze results and data from peer reviews
PLAN	1	PLAN 1.1	Develop a list of tasks
PLAN	1	PLAN 1.2	Assign people to tasks
PLAN	2	PLAN 2.1	Develop and keep updated the approach for accomplishing the work
PLAN	2	PLAN 2.2	Plan for the knowledge and skills needed to perform the work
PLAN	2	PLAN 2.3	Based on recorded estimates, develop and keep the budget and schedule updated
PLAN	2	PLAN 2.4	Plan the involvement of identified stakeholders
PLAN	2	PLAN 2.5	Plan transition to operations and support
PLAN	2	PLAN 2.6	Ensure plans are feasible by reconciling available and estimated resources
PLAN	2	PLAN 2.7	Develop the project plan, ensure consistency among its elements, and keep it updated
PLAN	2	PLAN 2.8	Review plans and obtain commitments from affected stakeholders
PLAN	3	PLAN 3.1	Use the organization's set of standard processes and tailoring guidelines to develop, keep updated, and follow the project process
PLAN	3	PLAN 3.2	Develop a plan and keep it updated, using the project process, the organization's process assets, and the measurement repository
PLAN	3	PLAN 3.3	Identify and negotiate critical dependencies
PLAN	3	PLAN 3.4	Plan for the project environment and keep it updated based on the organization's standards
PLAN	4	PLAN 4.1	Use statistical and other quantitative techniques to develop and keep the project processes updated to enable achievement of the quality and process performance objectives
PAD	1	PAD 1.1	Develop process assets to perform the work
PAD	2	PAD 2.1	Determine what process assets will be needed to perform the work
PAD	2	PAD 2.2	Develop, buy, or reuse process assets
PAD	2	PAD 2.3	Make processes and assets available
PAD	3	PAD 3.1	Develop, keep updated, and follow a strategy for building and updating process assets
PAD	3	PAD 3.2	Develop, record, and keep updated a process architecture that describes the structure of the organization's processes and process assets
PAD	3	PAD 3.3	Develop, keep updated, and make processes and assets available for use
PAD	3	PAD 3.4	Develop, keep updated, and use tailoring criteria and guidelines for the set of standard processes and assets
PAD	3	PAD 3.5	Develop, keep updated, and make the organization's process asset library available for use
PAD	3	PAD 3.6	Develop, keep updated, and make work environment standards available for use

PAD	3	PAD 3.7	Develop, keep updated, and make organizational measurement and analysis standards available for use
PCM	1	PCM 1.1	Develop a support structure to provide process guidance, identify and fix process problems, and continuously improve processes
PCM	1	PCM 1.2	Appraise the current process implementation and identify strengths and weaknesses
PCM	1	PCM 1.3	Address improvement opportunities or process issues
PCM	2	PCM 2.1	Identify improvements to the processes and process assets
PCM	2	PCM 2.2	Develop, keep updated, and follow plans for implementing selected process improvements
PCM	3	PCM 3.1	Develop, keep updated, and use process improvement objectives traceable to the business objectives
PCM	3	PCM 3.2	Identify processes that are the largest contributors to meeting business objectives
PCM	3	PCM 3.3	Explore and evaluate potential new processes, techniques, methods, and tools to identify improvement opportunities
PCM	3	PCM 3.4	Provide support for implementing, deploying, and sustaining process improvements
PCM	3	PCM 3.5	Deploy organizational standard processes and process assets
PCM	3	PCM 3.6	Evaluate the effectiveness of deployed improvements in achieving process improvement objectives
PCM	4	PCM 4.1	Use statistical and other quantitative techniques to validate selected performance improvements against proposed improvement expectations, business objectives, or quality and process performance objectives
PQA	1	PQA 1.1	Identify and address process and work product issues
PQA	2	PQA 2.1	Develop, keep updated, and follow a quality assurance approach and plan based on historical quality data
PQA	2	PQA 2.2	Throughout the project, objectively evaluate selected performed processes and work products against the recorded process and applicable standards
PQA	2	PQA 2.3	Communicate quality and non-compliance issues and ensure their resolution
PQA	2	PQA 2.4	Record and use results of quality assurance activities
PQA	3	PQA 3.1	Identify and record opportunities for improvement during quality assurance activities
PI	1	PI 1.1	Assemble solutions and deliver to the customer
PI	2	PI 2.1	Develop, keep updated, and follow an integration strategy

PI	2	PI 2.2	Develop, keep updated, and use the integration environment
PI	2	PI 2.3	Develop, keep updated, and follow procedures and criteria for integrating solutions and components
PI	2	PI 2.4	Confirm, prior to integration, that each component has been properly identified and operates according to its requirements and design
PI	2	PI 2.5	Evaluate integrated components to ensure conformance to the solution's requirements and design
PI	2	PI 2.6	Integrate solutions and components according to the integration strategy
PI	3	PI 3.1	Review and keep updated interface or connection descriptions for coverage, completeness, and consistency throughout the solution's life
PI	3	PI 3.2	Confirm, prior to integration, that component interfaces or connections comply with interface or connection descriptions
PI	3	PI 3.3	Evaluate integrated components for interface or connection compatibility
PI	1	PI 1.1	Assemble solutions and deliver to the customer
PI	2	PI 2.1	Develop, keep updated, and follow an integration strategy
PI	2	PI 2.2	Develop, keep updated, and use the integration environment
PI	2	PI 2.3	Develop, keep updated, and follow procedures and criteria for integrating solutions and components
PI	2	PI 2.4	Confirm, prior to integration, that each component has been properly identified and operates according to its requirements and design
PI	2	PI 2.5	Evaluate integrated components to ensure conformance to the solution's requirements and design
PI	2	PI 2.6	Integrate solutions and components according to the integration strategy
PI	3	PI 3.1	Review and keep updated interface or connection descriptions for coverage, completeness, and consistency throughout the solution's life
PI	3	PI 3.2	Confirm, prior to integration, that component interfaces or connections comply with interface or connection descriptions
PI	3	PI 3.3	Evaluate integrated components for interface or connection compatibility
RDM	1	RDM 1.1	Record requirements
RDM	2	RDM 2.1	Elicit stakeholder needs, expectations, constraints, and interfaces or connections
RDM	2	RDM 2.2	Transform stakeholder needs, expectations, constraints, and interfaces or connections into prioritized customer requirements
RDM	2	RDM 2.3	Develop an understanding with the requirements providers on the meaning of the requirements

RDM	2	RDM 2.4	Obtain commitment from project participants that they can implement the requirements
RDM	2	RDM 2.5	Develop, record, and maintain bidirectional traceability among requirements and activities or work products
RDM	2	RDM 2.6	Ensure that plans and activities or work products remain consistent with requirements
RDM	3	RDM 3.1	Develop and keep requirements updated for the solution and its components
RDM	3	RDM 3.2	Develop operational concepts and scenarios
RDM	3	RDM 3.3	Allocate the requirements to be implemented
RDM	3	RDM 3.4	Identify, develop, and keep updated interface or connection requirements
RDM	3	RDM 3.5	Ensure that requirements are necessary and sufficient
RDM	3	RDM 3.6	Balance stakeholder needs and constraints
RDM	3	RDM 3.7	Validate requirements to ensure the resulting solution will perform as intended in the target environment
RSK	1	RSK 1.1	Identify and record risks or opportunities and keep them updated
RSK	2	RSK 2.1	Analyze identified risks or opportunities
RSK	2	RSK 2.2	Monitor identified risks or opportunities and communicate status to affected stakeholders
RSK	3	RSK 3.1	Identify and use risk or opportunity categories
RSK	3	RSK 3.2	Define and use parameters for risk or opportunity analysis and handling
RSK	3	RSK 3.3	Develop and keep updated a risk or opportunity management strategy
RSK	3	RSK 3.4	Develop and keep updated risk or opportunity management plans
RSK	3	RSK 3.5	Manage risks or opportunities by implementing planned risk or opportunity management activities
SAM	1	SAM 1.1	Develop and record the supplier agreement
SAM	1	SAM 1.2	Accept or reject the supplier deliverables
SAM	1	SAM 1.3	Process supplier invoices
SAM	2	SAM 2.1	Monitor supplier as specified in the supplier agreement and keep agreement updated
SAM	2	SAM 2.2	Perform activities as specified in the supplier agreement
SAM	2	SAM 2.3	Verify that the supplier agreement is satisfied before accepting the acquired supplier deliverable
SAM	2	SAM 2.4	Manage invoices submitted by the supplier according to the supplier agreements
SAM	3	SAM 3.1	Select technical supplier deliverables for analysis and conduct technical reviews
SAM	3	SAM 3.2	Select and monitor supplier processes and deliverables based on criteria in the supplier agreement

SAM	4	SAM 4.1	Select measures and apply analytical techniques to quantitatively manage supplier performance to achieve quality and process performance objectives
TS	1	TS 1.1	Build a solution to meet requirements
TS	2	TS 2.1	Design and build a solution to meet requirements
TS	2	TS 2.2	Evaluate the design and address identified issues
TS	2	TS 2.3	Provide guidance on use of the solution
TS	3	TS 3.1	Develop criteria for design decisions
TS	3	TS 3.2	Develop alternative solutions for selected components
TS	3	TS 3.3	Perform a build, buy, or reuse analysis
TS	3	TS 3.4	Select solutions based on design criteria
TS	3	TS 3.5	Develop, keep updated, and use information needed to implement the design
TS	3	TS 3.6	Design solution interfaces or connections using established criteria
VV	1	VV 1.1	Perform verification to ensure the requirements are implemented and record and communicate results
VV	1	VV 1.2	Perform validation to ensure the solution will function as intended in its target environment and record and communicate results
VV	2	VV 2.1	Select components and methods for verification and validation
VV	2	VV 2.2	Develop, keep updated, and use the environment needed to support verification and validation
VV	2	VV 2.3	Develop, keep updated, and follow procedures for verification and validation
VV	3	VV 3.1	Develop, keep updated, and use criteria for verification and validation
VV	3	VV 3.2	Analyze and communicate verification and validation results

Anexo IV Matriz de relación áreas de prácticas -artefactos Scrum, para la definición de Requisitos funcionales

Los datos de esta tabla son una adaptación de (Henriquez et al., 2022)

Category Area	Capability Area	Madurity Level	CMMI-DEV V2.0 Practice Area	CMMI-DEV V2.0 Practice	Related Agile Practice
Doing	Ensuring Quality	Level 2	RDM	RDM 2.1	Backlog Grooming
Doing	Ensuring Quality	Level 2	RDM	RDM 2.1	User Stories
Doing	Ensuring Quality	Level 2	RDM	RDM 2.1	Epics
Doing	Ensuring Quality	Level 2	RDM	RDM 2.1	Test Driven Development
Doing	Ensuring Quality	Level 2	RDM	RDM 2.2	Backlog Grooming
Doing	Ensuring Quality	Level 2	RDM	RDM 2.2	User Stories
Doing	Ensuring Quality	Level 2	RDM	RDM 2.2	Epics
Doing	Ensuring Quality	Level 2	RDM	RDM 2.2	Test Driven Development
Doing	Ensuring Quality	Level 2	RDM	RDM 2.3	Sprint Planning
Doing	Ensuring Quality	Level 2	RDM	RDM 2.3	Backlog Grooming
Doing	Ensuring Quality	Level 2	RDM	RDM 2.3	Epics
Doing	Ensuring Quality	Level 2	RDM	RDM 2.3	Product Backlog
Doing	Ensuring Quality	Level 2	RDM	RDM 2.4	Sprint Planning
Doing	Ensuring Quality	Level 2	RDM	RDM 2.4	Backlog Grooming

Doing	Ensuring Quality	Level 2	RDM	RDM 2.4	Epics
Doing	Ensuring Quality	Level 2	RDM	RDM 2.4	Product Backlog
Doing	Ensuring Quality	Level 2	RDM	RDM 2.5	Backlog Grooming
Doing	Ensuring Quality	Level 2	RDM	RDM 2.5	Epics
Doing	Ensuring Quality	Level 2	RDM	RDM 2.5	Product Backlog
Doing	Ensuring Quality	Level 2	RDM	RDM 2.6	Release Planning
Doing	Ensuring Quality	Level 2	RDM	RDM 2.6	Backlog Grooming
Doing	Ensuring Quality	Level 2	RDM	RDM 2.6	Epics
Improving	Improving Performance	Level 2	MPM	MPM 2.3	Historical velocity
Improving	Improving Performance	Level 2	MPM	MPM 2.3	Burndown Charts
Improving	Improving Performance	Level 2	MPM	MPM 2.5	Historical velocity

