

Trabajo Fin de Grado

Grado en Ingeniería de Tecnologías Industriales

Aplicación práctica de la gestión de proyectos mediante la metodología Agile

Autor: Jaime Candau Otero

Tutor: Guillermo Montero Fernández - Vivancos

Dpto. Organización Industrial y Gestión de Empresas II
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2026



Trabajo Fin de Grado
Ingeniería de las Tecnologías Industriales

Aplicación práctica de la gestión de proyectos mediante la metodología Agile

Autor:
Jaime Candau Otero

Tutor:
Guillermo Montero Fernández - Vivancos
Profesor asociado

Dpto. Organización Industrial y Gestión de Empresas II
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2026

Trabajo Fin de Grado: Aplicación práctica de la gestión de proyectos mediante la metodología Agile

Autor: Jaime Candau Otero

Tutor: Guillermo Montero Fernández -Vivancos

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2026

El Secretario del Tribunal

A mi familia

A mis maestros

Agradecimientos

En primer lugar, quiero expresar mi más sincero agradecimiento a mi tutor, D. Guillermo Montero Fernández - Vivancos, por su compromiso e ilusión al guiarme en este nuevo camino que ha sido el descubrimiento de la metodología Agile. Gracias por su atención constante durante todo el proceso y por transmitirme esa visión innovadora de la gestión de proyectos, la cual ha sido fundamental para la realización de este Trabajo de Fin de Grado.

A mis amigos y compañeros de carrera, con quienes he compartido este largo trayecto. Gracias por todas esas experiencias inolvidables que, más allá de lo académico, nos servirán para seguir formándonos en un futuro y por convertir estos años en una etapa única de mi vida.

De manera muy especial, quiero agradecer a mi familia, el pilar fundamental que me ha permitido aguantar la presión de una carrera tan compleja como es Industriales. Sin vuestro apoyo incondicional, este Trabajo de Fin de Grado no sería hoy una realidad. A mi madre y a mi padre, gracias por aguantarme día a día, por vuestra paciencia infinita y por ser quienes me han dado ánimos y fuerza cuando mi cuerpo y mi mente decían basta. Vuestro ejemplo de esfuerzo y dedicación es lo que me ha traído hasta aquí.

A todos, gracias de corazón.

Jaime Candau Otero

Sevilla, 2026

Resumen

La gestión de proyectos en la ingeniería industrial se ha apoyado tradicionalmente en el modelo en cascada (Waterfall): planificarlo todo de principio a fin antes de empezar. Sin embargo, en un entorno donde la tecnología, los costes y las necesidades del cliente cambian constantemente, esta rigidez se ha convertido en un problema. La metodología Agile responde a este reto dividiendo el proyecto en ciclos cortos de trabajo que permiten validar resultados y reducir el riesgo de forma continua.

El objetivo de este Trabajo de Fin de Grado es demostrar que la combinación de Agile y la Inteligencia Artificial constituye una herramienta capaz de modernizar la gestión de proyectos industriales. Para ello, tras sentar las bases teóricas de Scrum y Kanban, he diseñado y configurado cinco asistentes personalizados (Gemas) sobre el modelo Gemini, cada uno orientado a una fase concreta del ciclo de vida del proyecto: la generación de Historias de Usuario, el refinamiento del backlog, la detección temprana de riesgos, el cálculo de métricas de rendimiento y la gestión de solicitudes de cambio.

Cada herramienta se ha validado mediante casos de aplicación que reflejan situaciones reales del entorno industrial, algunos de ellos inspirados en mi experiencia durante las prácticas en Accenture. Los resultados muestran que la IA, correctamente dirigida mediante ingeniería de prompts, libera al ingeniero del trabajo administrativo repetitivo y eleva el estándar de calidad de la documentación, permitiéndole centrarse en lo que realmente aporta valor: la toma de decisiones técnicas.

Abstract

Project management in industrial engineering has traditionally relied on the Waterfall model: planning everything from start to finish before any work begins. However, in an environment where technology, costs and customer needs change constantly, this rigidity has become a problem. The Agile methodology addresses this challenge by breaking projects into short work cycles that allow results to be validated and risk to be reduced continuously.

The aim of this Bachelor's Thesis is to demonstrate that combining Agile with Artificial Intelligence provides a powerful tool for modernising the management of industrial projects. After establishing the theoretical foundations of Scrum and Kanban, I designed and configured five custom assistants (Gems) built on the Gemini model, each targeting a specific stage of the project life cycle: generating User Stories, refining the backlog, detecting risks early, calculating performance metrics and managing change requests.

Each tool was validated through application cases reflecting real industrial situations, some of them inspired by my experience during my internship at Accenture. The results show that AI, when properly guided through prompt engineering, frees engineers from repetitive administrative work and raises the quality standard of project documentation, allowing them to focus on what truly adds value: technical decision-making.

Índice

Agradecimientos	9
Resumen	11
Abstract	13
Índice	15
Índice de tablas	18
Índice de figuras	20
1 Objetivos y alcance	22
2 Metodología Agile	24
2.1 <i>¿Qué es la metodología Agile?</i>	24
2.1.1 Contexto histórico	26
2.2 <i>El marco de trabajo Scrum: roles, artefactos e integración de la IA</i>	27
2.2.1 Los tres roles principales	27
2.2.2 Daily Scrum	28
2.2.3 Artefactos: la base de la transparencia	29
2.2.4 Scrum e inteligencia artificial	30
2.3 <i>Historias de usuario</i>	31
2.3.1 Las 3 "C" de las historias de usuario	31
2.3.2 Criterios INVEST para ingenieros	31
2.4 <i>Otros marcos de trabajos con metodología Agile</i>	32
2.4.1 Kanban	32
3 Inteligencia artificial aplicada a la gestión de proyectos	35
3.1 <i>De la IA clásica a los modelos de lenguaje</i>	35
3.2 <i>Ingeniería de prompts: cómo se le habla a la máquina</i>	35
3.3 <i>Agentes y asistentes especializados</i>	36
4 Aplicación de la IA en Agile: gemas e historias de usuario	38
4.1 <i>Creación de la primera "Gema"</i>	38
4.2 <i>El cerebro de la Gema</i>	39
4.2.1 Definición del rol y contexto	39
4.2.2 Reglas de configuración	39
4.3 <i>Aplicación real</i>	40
4.3.1 Input	40
4.3.2 Prompts en acción	40
4.3.3 Análisis de resultados	41
4.4 <i>Creación de la segunda Gema</i>	42
4.4.1 Definición de rol y contexto	42
4.4.2 Reglas de configuración	42
4.5 <i>Aplicación real</i>	43
4.5.1 Estudio de resultados	45

5	Aplicación de la IA en Agile: Gemas para control y métricas de Sprint	47
5.1	<i>Creación de la primera Gema</i>	47
5.1.1	Aplicación real	48
5.2	<i>Creación de la segunda Gema</i>	50
5.2.1	Aplicación real	50
6	Aplicación de la IA en Agile: gemas y gestión de solicitudes de cambio (Change Requests)	55
6.1	<i>Reglas de configuración</i>	55
6.2	<i>El cerebro de la Gema</i>	56
6.3	<i>Aplicación real</i>	56
6.3.1	Contexto del negocio:	56
6.3.2	Contexto del Sprint y estado actual:	56
6.3.3	La solicitud de cambio (Change Request):	57
6.3.4	Datos técnicos extra para el análisis:	57
6.4	<i>Respuesta de la Gema</i>	57
6.4.1	Estrategias de mitigación prescriptivas	58
6.4.2	Borrador de correo electrónico	59
7	Análisis comparativo de las Gemas desarrolladas	61
7.1	<i>Interpretación gráfica de la comparación</i>	62
8	Conclusión	65
	Referencias	68

ÍNDICE DE TABLAS

Tabla 1: Excel seguimiento Daily	29
Tabla 2: Gema 2 fase 1	44
Tabla 3: Gema Radar de Riesgos	49
Tabla 4: Analista KPI Fase 1	51
Tabla 5: Analista KPI 2 caso Gema Fase 1	52
Tabla 6: Respuesta Gema Change Request	58
Tabla 7: Comparativa de Gemas	61

ÍNDICE DE FIGURAS

Figura 1: Modelo en cascada	25
Figura 2: Sprint	26
Figura 3: Roles Scrum	28
Figura 4: Artefactos Scrum	30
Figura 5: Scrum e IA	30
Figura 6: Historia de usuario	32
Figura 7. Kanban	33
Figura 8: Gema Agile 1	39
Figura 9: Respuesta Gema 1.1	41
Figura 10: Respuesta Gema 1.2	41
Figura 11: Segunda Gema	42
Figura 12: Escala de Fibonacci	43
Figura 13: Respuesta fase 1 primera Gema riesgo	48
Figura 14: Respuesta Gema Change Request	58
Figura 15: Perfil comparativo de las cinco Gemas según los seis criterios de evaluación	63

1 OBJETIVOS Y ALCANCE

La ingeniería industrial se ha apoyado históricamente en metodologías predictivas rígidas que, aunque funcionan bien en entornos estables, fallan cuando hay que gestionar la incertidumbre de los proyectos actuales. El objetivo principal de este Trabajo de Fin de Grado es demostrar que la metodología Agile, lejos de ser exclusiva del desarrollo de software, es un marco de trabajo válido y útil para la organización industrial, especialmente cuando se apoya en la Inteligencia Artificial.

En concreto, se busca probar cómo la configuración de asistentes virtuales especializados (Gemas) mediante ingeniería de *prompts* permite automatizar la carga administrativa, reducir riesgos y devolver al ingeniero el tiempo necesario para centrarse en lo que de verdad importa: la toma de decisiones técnicas.

Para alcanzar este fin, el proyecto se estructura en torno a los siguientes objetivos específicos:

- **Analizar la base teórica:** comparar el modelo tradicional en cascada (*Waterfall*) con los marcos ágiles (Scrum y Kanban), justificando por qué la iteración tiene sentido en entornos industriales.
- **Diseñar herramientas de IA aplicadas:** configurar cinco asistentes basados en Inteligencia Artificial (Gemas), cada uno enfocado en un punto concreto donde la gestión del proyecto suele atascarse: la redacción de historias de usuario, el refinamiento técnico, la detección temprana de riesgos, la revisión de métricas y la gestión de solicitudes de cambio.
- **Validar mediante casos de estudio:** someter cada herramienta a situaciones que reproduzcan la realidad de una planta o de una oficina técnica, evaluando la utilidad y la calidad técnica de las respuestas que genera la IA.

En cuanto al alcance, este trabajo se centra exclusivamente en la capa de gestión y de ingeniería de proyectos. No se pretende programar un modelo de Inteligencia Artificial desde cero ni desarrollar un software comercial, sino aprovechar los Modelos de Lenguaje de Gran Escala (LLM) actuales, en concreto Gemini, configurándolos para que trabajen siguiendo los estándares técnicos de la industria. Los casos prácticos presentados cubren el ciclo de vida completo de un Sprint, desde la llegada de unos requisitos iniciales poco definidos hasta el diagnóstico final de rendimiento.

2 METODOLOGÍA AGILE

Hasta ahora, en la carrera nos han enseñado que para gestionar un proyecto industrial hay que planificarlo todo de principio a fin antes de empezar, como si fuera una línea recta. Esto es lo que se conoce como modelo en cascada o “Waterfall”. El problema es que hoy en día, con lo rápido que cambia todo, esa forma de trabajar se queda rígida. Si tardas seis meses en planificar y luego algo cambia, tu plan ya no sirve para nada.

Aquí es donde entra la *metodología Agile* [1]. No se trata de "hacer las cosas más rápido" sin más, sino de trabajar de una forma distinta: dividiendo el proyecto en trozos pequeños y entregando resultados cada poco tiempo. Así, si te equivocas, te das cuenta a las dos semanas y no al final del año. En este Trabajo de Fin de Grado quiero demostrar cómo esto se puede aplicar a la organización industrial, que es un sector que tradicionalmente ha sido muy cuadrulado.

Además, hoy tenemos una ventaja que no existía hace unos años: la Inteligencia Artificial. En este trabajo voy a explicar cómo usar la IA (específicamente a través de *prompts*) para que nos ayude a redactar las tareas del equipo o las llamadas "Historias de Usuario". Básicamente, la IA nos quita el trabajo aburrido de oficina para que nosotros, como ingenieros, nos centremos en tomar las decisiones importantes. En resumen, Agile y la IA son el combo perfecto para que la gestión de proyectos deje de ser un dolor de cabeza burocrático y pase a ser algo dinámico y eficiente.

2.1 ¿Qué es la metodología Agile?

Para entender qué es Agile, primero hay que entender de dónde venimos. En ingeniería, lo "normal" siempre ha sido usar el modelo en cascada o *Waterfall*. Es el sistema que se ha usado para construir puentes o fábricas durante décadas: primero diseñas todo, luego compras materiales, luego construyes y al final entregas. Este modelo merece todo nuestro respeto porque es súper ordenado y, si el proyecto está clarísimo desde el minuto uno, funciona muy bien. El problema es que hoy en día, en un mundo donde la tecnología y los costes cambian cada semana debido a su constante mejoría, intentar planificar a dos años vista se ha convertido en un reto difícilmente asumible.

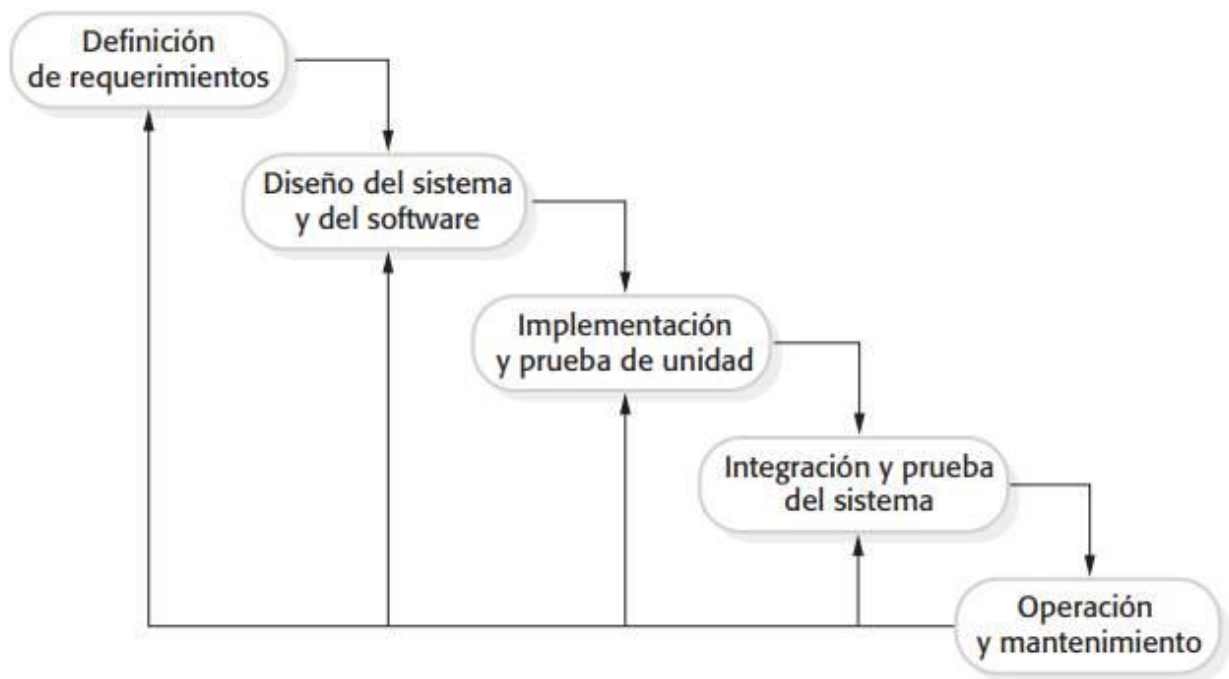


Figura 1: Modelo en cascada

Agile es dejar de ver un proyecto como una "gran entrega final" y empezar a verlo como una serie de pequeñas entregas constantes. En lugar de estar encerrado en la oficina técnica seis meses diseñando una máquina perfecta para luego descubrir que no cabe en la planta, con Agile diseñas una parte, la pruebas, recibes *feedback* y sigues con la siguiente. Es decir, nos centramos en pequeñas mejoras constantes que nos ayudan a alcanzar un resultado final de una manera más seguida. Esta filosofía se integra junto a los enfoques predictivos clásicos descritos por el PMI [4], y ha sido extendida específicamente al ámbito de la gestión de proyectos por Highsmith [16].

Es una forma de gestionar la incertidumbre. En ingeniería industrial, esto significa que aceptamos que no lo sabemos todo desde el principio. Usamos ciclos cortos de trabajo para ir reduciendo el riesgo poco a poco. Así, si hay que cambiar algo (porque ha subido el precio del acero o porque el cliente ha cambiado de idea), no tenemos que tirar a la basura meses de trabajo, sino solo el trabajo de las últimas dos semanas. Es una manera de ir dejando pequeños *Checkpoints* en nuestro proyecto.

Esta metodología Agile es posible gracias a un término que conocemos como Sprints. El Sprint es la unidad básica sobre la que se asienta toda la metodología. Se define como un intervalo de tiempo prefijado (normalmente entre una y cuatro semanas) durante el cual el equipo trabaja para crear un incremento de valor real. La característica principal del Sprint es su naturaleza protegida: una vez que comienza, no se pueden realizar cambios que afecten al objetivo marcado, lo que garantiza el foco y la estabilidad del equipo.

El funcionamiento del Sprint se basa en un ciclo de inspección y adaptación constante que consta de los siguientes eventos formales:

1. Planificación del Sprint (Sprint Planning): Es la fase inicial donde el equipo selecciona los elementos prioritarios del *Product Backlog* (lista de requisitos) y define el objetivo del ciclo. Se decide qué se va a entregar y cómo se va a realizar el trabajo técnico necesario.
2. Scrum Diario (Daily Scrum): Es un evento de sincronización de 15 minutos que se realiza cada 24 horas. Su objetivo es evaluar el progreso hacia el objetivo del Sprint y ajustar la planificación del día, identificando posibles impedimentos que bloqueen el flujo de trabajo.
3. Revisión del Sprint (Sprint Review): Se lleva a cabo al finalizar el ciclo para inspeccionar el resultado obtenido. En esta sesión, el equipo y los interesados analizan qué se ha completado y qué no, permitiendo ajustar el *Product Backlog* para los siguientes ciclos según la situación real del proyecto.

4. Retrospectiva del Sprint (Sprint Retrospective): Es el último paso del ciclo y se centra en la mejora del proceso. El equipo analiza su propia dinámica de trabajo, las herramientas utilizadas y las relaciones personales, con el fin de identificar mejoras concretas que se aplicarán de inmediato en el siguiente Sprint.

Este ciclo permite que la gestión de proyectos industriales pase de ser un proceso rígido a uno incremental, donde el riesgo se minimiza al validar resultados en periodos de tiempo muy cortos.

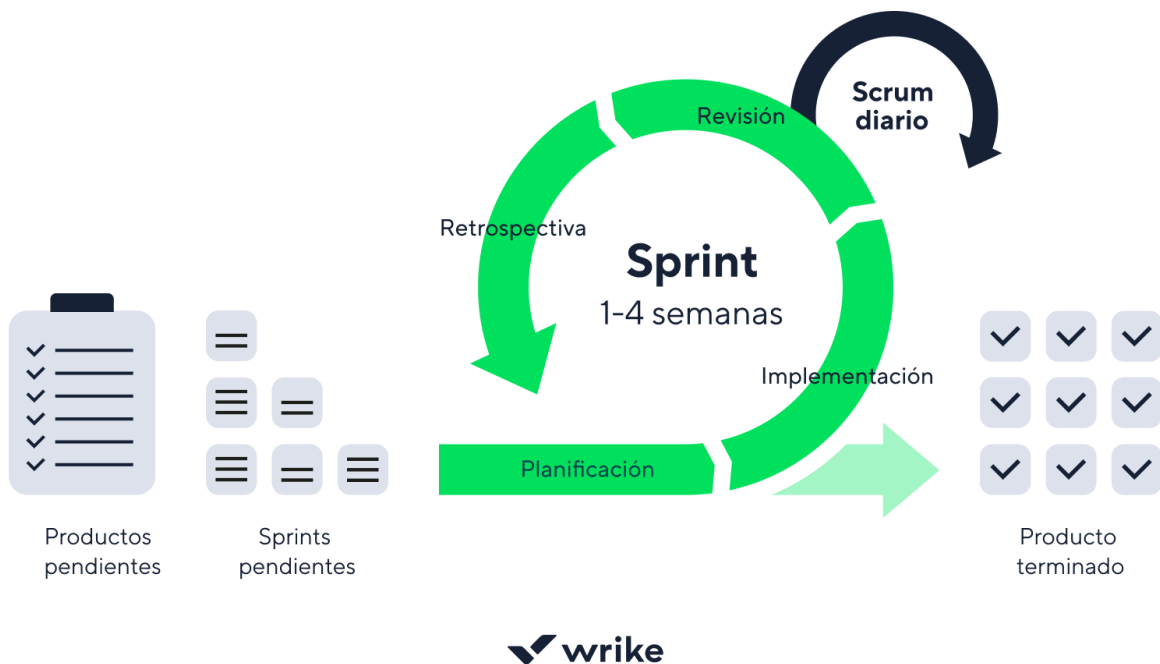


Figura 2: Sprint

2.1.1 Contexto histórico

Si queremos ser precisos, el salto real de la filosofía de Toyota [8], cuyo impacto global describieron Womack, Jones y Roos [11] a lo que hoy llamamos Agile ocurrió gracias a un artículo de 1986 llamado *“The New New Product Development Game”* [2]. En él, dos japoneses (Takeuchi y Nonaka) compararon la forma de trabajar de las mejores empresas del mundo con el Rugby.

Explicaban que los métodos tradicionales de ingeniería eran como una carrera de relevos: uno corre, termina su parte y le pasa el testigo al siguiente. El problema es que, si al primero se le cae el testigo, el último ya no gana. Ellos propusieron que los equipos de ingeniería trabajaran como un equipo de rugby: todos avanzan juntos, pasándose el balón de lado a lado para llegar a la meta. De aquí viene precisamente el nombre de Scrum (que significa 'melé' en inglés).

Este enfoque caló hondo porque demostró que, en empresas como Honda o Canon, los equipos que trabajaban de forma multidisciplinar desde el inicio terminaban antes y con menos errores que los que seguían el orden lógico de toda la vida. Años más tarde, los principios Lean nacidos en la industria se trasladaron al desarrollo software gracias al trabajo de Mary y Tom Poppendieck [14], cerrando así el puente entre la filosofía de fabricación de Toyota y los marcos ágiles modernos.

2.2 El marco de trabajo Scrum: roles, artefactos e integración de la IA

Como se ha mencionado anteriormente al hablar del contexto histórico, el término Scrum proviene del mundo del rugby, haciendo referencia a la "melé": ese momento en el que el equipo trabaja como un bloque compacto para avanzar posiciones. Si Agile es la filosofía, Scrum es el marco de trabajo que nos da las reglas del juego [3], [15]. Para un ingeniero industrial, Scrum es la herramienta que permite sustituir el mando y control tradicional por un sistema de autoorganización, donde el equipo es el responsable de gestionar su propio trabajo.

En la actualidad, Scrum no solo se apoya en el talento humano, sino que se ha convertido en el entorno ideal para integrar la Inteligencia Artificial [17]. La estructura de Scrum permite que la IA actúe como un acelerador en la definición de tareas y en el análisis de datos, convirtiéndose casi en un "asistente invisible" que optimiza cada parte del proceso que veremos a continuación.

2.2.1 Los tres roles principales

Para que esta "melé" funcione, Scrum define tres roles con responsabilidades muy claras. No son jerarquías, sino funciones distintas para que el proyecto no se detenga:

1. **Product Owner:** Es la persona que tiene la visión de qué es lo que hay que construir y por qué. Su función principal es maximizar el valor del trabajo que hace el equipo. Es el responsable de gestionar la lista de tareas. En este rol, la IA es clave para analizar tendencias del mercado o de la planta de producción y ayudar a priorizar qué tareas son más urgentes.
2. **Scrum Master:** No es un jefe de proyecto tradicional, sino un "líder servicial". Su trabajo es asegurar que todo el equipo entiende y aplica Scrum, eliminando cualquier impedimento que les frene. Es quien facilita las reuniones y protege al equipo de interrupciones externas.
3. **Developers:** Es el grupo de profesionales que realizan el trabajo técnico. Son multidisciplinares y autoorganizados; nadie les dice "cómo" hacer su trabajo, ellos deciden la mejor forma técnica de alcanzar el objetivo del Sprint.

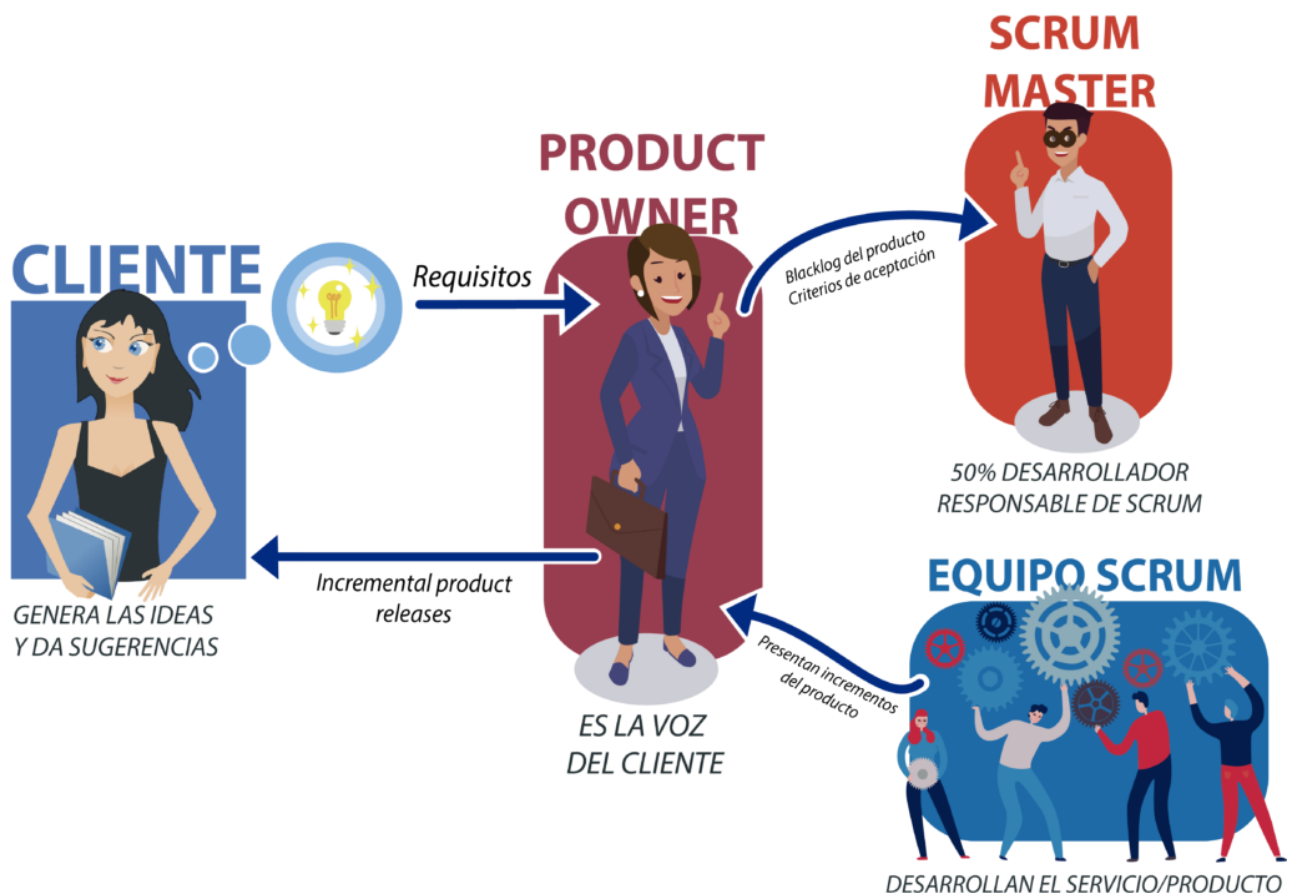


Figura 3: Roles Scrum

2.2.2 Daily Scrum

Dentro de la rutina de Scrum, la Daily Scrum (o reunión diaria) es probablemente la herramienta más potente para que la transparencia y la colaboración no se queden solo en el olvido. No es una reunión para darle explicaciones a un jefe, sino un espacio para que el equipo de desarrollo se ponga al día de avances y tareas a realizar dicho día. Aunque el Scrum Master la facilita y el Product Owner puede ir a escuchar, los protagonistas absolutos son los que están ejecutando el trabajo.

Para que esto no se convierta en una pérdida de tiempo, se siguen unas reglas muy claras: se hace siempre a la misma hora para crear un hábito y, lo más importante, tiene un tiempo máximo teórico de unos 15 minutos. Un truco clásico en las oficinas técnicas es hacerla de pie; parece una tontería, pero ayuda muchísimo a que la gente vaya al grano y no se disperse. En mi caso, en las que he tenido personalmente, las hacemos vía Teams con 15 minutos de llamada y con un Excel que tenemos preparado para perder el menor tiempo posible.

El objetivo es responder a tres preguntas que ponen a todo el mundo en la misma página:

1. **¿Qué hice ayer?** Para actualizar el progreso real.
2. **¿Qué voy a hacer hoy?** Para confirmar que seguimos alineados con los objetivos del Sprint.
3. **¿Tengo algún obstáculo?** Aquí es donde sale a la luz cualquier problema que esté frenando el avance.

Durante la reunión, el Sprint Backlog es nuestra brújula. Cada miembro actualiza el estado de sus tareas (pendiente, en curso o terminada), lo que permite ver de un vistazo si vamos por buen camino o si hay que reajustar algo. Si alguien termina antes, el Scrum Master ayuda a redistribuir el esfuerzo para que el equipo sea lo más eficiente posible.

En el fondo, la Daily no es solo "pasar información". Es la oportunidad diaria para que el equipo funcione como una unidad, eliminando bloqueos gracias al apoyo del Scrum Master y asegurando que, pase lo que pase, el foco

siempre esté en cumplir lo prometido en el Sprint. Al final, lo que ganamos es transparencia total, menos errores por falta de comunicación y una capacidad de reacción que el modelo tradicional de reuniones semanales simplemente no puede ofrecer.

Tabla 1: Excel seguimiento Daily

ID TAREA	TITULO	ESCENARIO	DESCRIPCIÓN	ACTUALIZACION	PRIORIDAD	STATUS	NEXT ACTION
ID_00X	Gestión de cambios y configuración de XXX	XXX	XXX		4	En estudio	Funcional
ID_0XX	Lanzar WFDocumentos desde WF ECN grande	XXX	XXX	18/1X/202X: XXX 04/1X/202X: XXX 06/1X/202X: XXX 10/0X/202X: XXX • Están los objetos aprobados en solución? • XXX lo está mirado Se ha abierto un IR	3	En estudio	Funcional
ID_0X0	Actualizar visios en DevOps	XXX	XXX	XXX	3	En estudio	Funcional
ID_0X0	Faltan Unidades atributos clasificación	XXX	XXX	XXX	2	En espera otros equipos	SAP
ID_0XX	XXX	XXX	XXX	XXX	2	Pendiente de IR	Funcional

2.2.3 Artefactos: la base de la transparencia

Scrum utiliza tres herramientas físicas para que todo el mundo sepa en qué punto está el proyecto en todo momento:

- **Product Backlog:** Es la lista maestra de todo lo que el producto o proyecto necesita. Es dinámica y nunca está cerrada. Aquí es donde los prompts de IA son más útiles, ayudando a desglosar grandes ideas en tareas pequeñas y manejables.
- **Sprint Backlog:** Es el conjunto de tareas elegidas del Product Backlog para ser realizadas durante el Sprint actual. Es el plan de acción inmediato.
- **Incremento:** Es la suma de todas las tareas completadas durante un Sprint que ya están listas para ser utilizadas o entregadas. Es el valor real y tangible que hemos generado.

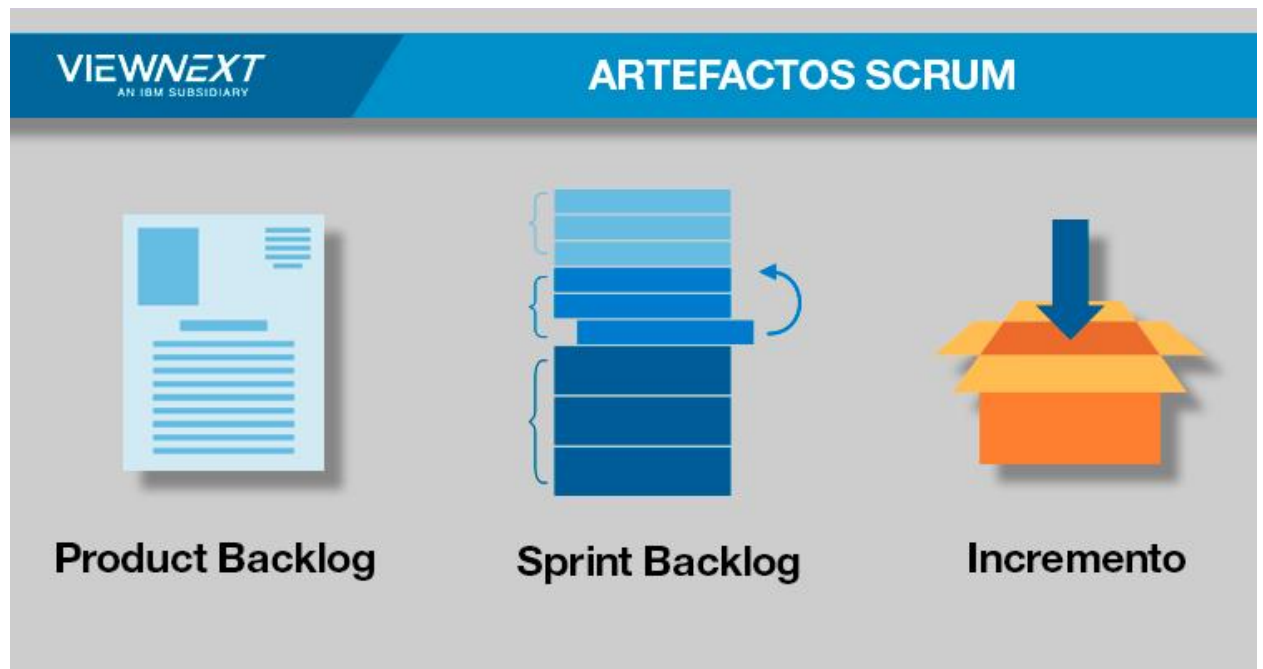


Figura 4: Artefactos Scrum

2.2.4 Scrum e inteligencia artificial

La implementación de Scrum en la organización industrial moderna no se puede entender sin la IA. Al ser un marco basado en ciclos cortos y feedback constante, genera una gran cantidad de datos que la IA puede procesar para predecir retrasos o sugerir mejoras en el flujo de trabajo. Como veremos más adelante, el uso de asistentes inteligentes en la redacción de estas tareas o en el seguimiento de los objetivos permite que los roles de Scrum sean mucho más eficientes y menos administrativos.



Figura 5: Scrum e IA

2.3 Historias de usuario

En la gestión tradicional, los requisitos se escribían en acumulaciones de cientos de páginas técnicas que casi nadie leía enteros. En Agile, utilizamos las Historias de Usuario [5]. Una Historia de Usuario es una explicación corta y sencilla de una funcionalidad, escrita desde la perspectiva que necesita mejorar el usuario final, ya sea un operador, un cliente o el director de la fábrica.

Su estructura suele seguir siempre este esquema: "Como [rol], quiero [acción], para que [beneficio]". Siempre la idea es transmitir un mensaje sencillo y preciso que sea entendible sin mayores complicaciones. Además, también se sugiere que todas sigan misma estructura cuando se suben para que queden guardadas de manera ordenada.

Por ejemplo, actualmente me encuentro realizando prácticas en Accenture, compañía líder en su sector que me está ayudando mucho a formarme tanto profesional como personalmente. En este proyecto que me encuentro involucrado, utilizamos estas historias de usuario de manera continua, subiendo indicaciones continuamente en DevOps de manera que queda todo organizado correctamente. (en una planta industrial, en lugar de poner "Instalar sensor de temperatura", pondríamos: *"Como operario de mantenimiento, quiero recibir una alerta en mi móvil cuando la máquina supere los 80°C, para evitar una parada por avería"*. Esta forma de escribir obliga al equipo a no perder de vista el valor real de lo que están haciendo.)

2.3.1 Las 3 "C" de las historias de usuario

Para que una Historia de Usuario sea competente, no basta con escribirla, sino que tiene que cumplir con el modelo de las 3 "C", formulado por Jeffries en el contexto de Extreme Programming [6], [13]:

1. Card (Tarjeta): La historia debe ser lo suficientemente breve como para caber en una tarjeta pequeña (Normalmente es un cuadro pequeño de texto colocado en la parte superior). Es el recordatorio de la necesidad.
2. Conversation: Lo más importante no es lo que está escrito, sino la posibilidad de comunicación entre el *Product Owner* y el equipo para aclarar los detalles técnicos. Aquí se puede dejar claro que el equipo realmente ha entendido el mensaje y los avances que se van teniendo.
3. Confirmation: Son los criterios de aceptación. Es decir, la lista de comprobación para saber que la tarea está realmente terminada y funciona. Una vez realizada la Historia de Usuario, puede quedar en Resolved, Accepted, In progress, ...

2.3.2 Criterios INVEST para ingenieros

Como ingenieros, estamos acostumbrados a que todo tenga una norma o un estándar de calidad. Para las Historias de Usuario, ese estándar es el acrónimo INVEST, propuesto por Bill Wake en 2003 [7]. Si una historia no cumple estos seis puntos, lo más probable es que nos dé problemas durante el Sprint.

La Palabra INVEST viene de:

- I de Independiente (Independent): Las historias no deberían estar "acopladas" entre sí. Si para terminar la tarea A necesito que la tarea B esté lista al 100%, tenemos un problema de flujo. En la medida de lo posible, cada historia debe poder completarse por separado para no bloquear al equipo. De otra manera se podría ver como que sigue un modelo de PUSH más que de PULL.
- N de Negociable (Negotiable): Una Historia de Usuario no es un contrato cerrado ni un plano inamovible. Es una invitación a la conversación. Debe dejar espacio para que el equipo discuta la mejor solución técnica con el *Product Owner*. Por ello la conversación entre equipo y Product Owner es crucial.
- V de Valiosa (Valuable): Cada tarea debe aportar un valor real al cliente o al proceso. Si escribimos una historia que no sirve para mejorar nada o que no aporta beneficio, es "desperdicio" (lo que en Lean

llamaríamos *Despilfarro* [10]) y deberíamos eliminarla.

- E de Estimable (Estimable): El equipo tiene que ser capaz de prever cuánto esfuerzo va a llevar. Si una historia es tan vaga que no sabemos si tardaremos dos horas o dos semanas, no podemos planificar el Sprint con seguridad.
- S de Pequeña (Small): Una historia debe ser lo suficientemente pequeña como para terminarse dentro de un solo Sprint. Si es demasiado grande (lo que llamamos una "Épica"), hay que trocearla en trozos más manejables.
- T de Testable (Testable): Tenemos que poder comprobar que funciona. Si no hay una forma clara de decir "esto está bien hecho" (mediante los criterios de aceptación que vimos en la Confirmación), entonces la historia no está bien definida.

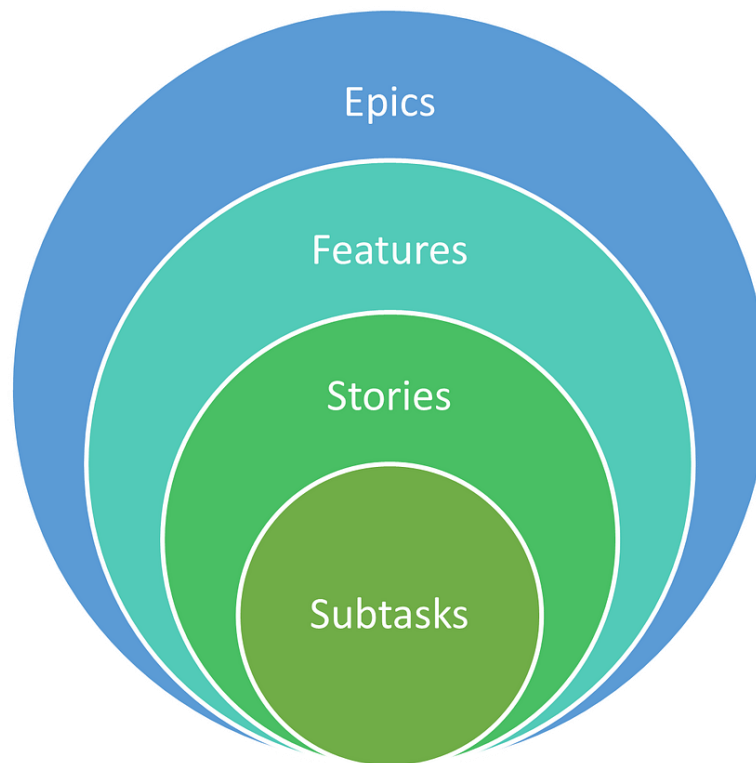


Figura 6: Historia de usuario

2.4 Otros marcos de trabajos con metodología Agile

A pesar de que Scrum es sin duda el mayor protagonista cuando hablamos de metodologías ágiles, la realidad es que en la industria existen otros métodos que, a veces, encajan mejor con el día a día de una planta o un departamento de ingeniería. No siempre necesitamos trabajar basándonos en Sprints cerrados, a veces lo que necesitamos es que el trabajo fluya sin interrupciones.

En este apartado vamos a ver las metodologías más usadas que complementan o sirven de alternativa a lo que hemos visto hasta ahora:

2.4.1 Kanban

Aunque solemos pensar en Kanban como un simple tablero visual, su origen en las fábricas de Toyota con Taiichi Ohno [8] nos enseña algo que no siempre es percibido por la mayoría: para ir más rápido, a veces hay que obligarse a ir más despacio. Kanban no es una metodología de "hacer mucho", sino de gestionar el flujo. Su enfoque es incremental y evolutivo, lo que significa que no necesitamos revolucionar la empresa de un día para

otro, sino mejorar el proceso paso a paso.

Lo que realmente desafía la gestión tradicional es que Kanban se basa en limitar el trabajo para aumentar la velocidad. Se apoya en valores como la garantía de calidad (no se pasa de fase si algo está mal), la reducción de despilfarros (eliminar lo que no aporta valor), la mejora continua (*Kaizen*) [12] y una flexibilidad total para pivotar cuando las prioridades cambian.

Para que este sistema funcione en la organización industrial, nos regimos por tres reglas de oro que cambian nuestra forma de ver el trabajo:

- **Visualizar el proceso:** No basta con saber qué hay que hacer; todo el equipo y los interesados deben ver el flujo en tiempo real. Esto no es solo para "informar", sino para que los problemas y cuellos de botella "salten a la vista" antes de que sea demasiado tarde. Mejora la comunicación porque el tablero habla por sí solo.
- **Limitar el trabajo en curso (WIP - Work In Progress) [9]:** Esta es la regla más contundente. Hay que fijar un número máximo de tareas que pueden estar en cada fase. Si una columna está llena, nadie puede meter más trabajo ahí. Esto nos obliga a colaborar: si hay un cuello de botella, el equipo se detiene para ayudar a desbloquearlo en lugar de seguir acumulando tareas pendientes que nadie termina. Esto se ve claramente en la imagen de abajo
- **Optimización del flujo de trabajo:** Aquí encontramos la parte más técnica. Medimos el Tiempo de Ciclo (lo que tardamos en terminar una tarea) y buscamos maximizar el Throughput (nuestro rendimiento real). El objetivo no es que la gente esté "ocupada", sino que las tareas se muevan de principio a fin de la manera más rápida y constante posible.

En definitiva, Kanban nos enseña que el despilfarro de recursos y el tiempo de inactividad no se combaten trabajando más horas, sino gestionando mejor los límites de nuestro sistema para minimizar errores y retrasos en la entrega.



Figura 7.Kanban

3 INTELIGENCIA ARTIFICIAL APLICADA A LA GESTIÓN DE PROYECTOS

3.1 De la IA clásica a los modelos de lenguaje

Hablar hoy de Inteligencia Artificial es entrar en un terreno donde el término se ha vuelto tan amplio que casi pierde significado. Conviene, por tanto, acotarlo antes de avanzar. La IA es, en su definición más extendida, la disciplina que se ocupa de construir sistemas capaces de realizar tareas que tradicionalmente requerían inteligencia humana, como razonar, aprender o tomar decisiones [18]. Dentro de este paraguas conviven enfoques muy distintos, desde los sistemas expertos basados en reglas hasta los modelos estadísticos del Machine Learning, pero la rama que ha transformado el panorama en los últimos años, y la que vertebra este trabajo, es la del Deep Learning y, en concreto, los modelos de lenguaje a gran escala o LLM (Large Language Models).

Un LLM es, en esencia, un modelo estadístico entrenado sobre cantidades masivas de texto que aprende a predecir la palabra siguiente más probable a partir de un contexto dado. La arquitectura que hizo posible este salto cualitativo es el Transformer, propuesta por Vaswani et al. en 2017 [19], que introdujo el mecanismo de atención (self-attention) y rompió las limitaciones de las redes recurrentes anteriores. Modelos como GPT, Gemini o Claude son todos descendientes directos de esa arquitectura. Lo verdaderamente relevante desde el punto de vista práctico no es la complejidad matemática que hay detrás, sino una propiedad emergente: cuando un modelo de este tipo alcanza cierto tamaño, deja de limitarse a completar texto y empieza a mostrar capacidades que no estaban explícitamente programadas, como resumir, traducir, razonar de forma encadenada o seguir instrucciones complejas [20].

Conviene matizar, sin embargo, que un LLM no "entiende" en el sentido humano del término. Funciona por correlación estadística sobre su corpus de entrenamiento, lo que implica que puede generar respuestas plausibles pero incorrectas, un fenómeno conocido como alucinación. Esta limitación es lo que justifica todo lo que viene después en este trabajo: si la IA va a tocar un proyecto industrial, no puede dejarse suelta. Necesita restricciones, contexto y control humano.

3.2 Ingeniería de prompts: cómo se le habla a la máquina

El comportamiento de un LLM depende, en proporciones similares, de dos cosas: del modelo en sí y de cómo se le da la instrucción. A esta segunda disciplina se la conoce como ingeniería de prompts (prompt engineering), y consiste en diseñar las indicaciones que recibe el modelo para que produzca salidas útiles, precisas y consistentes [21]. Es una práctica más cercana al diseño que a la programación clásica: no se escribe código, se escribe una guía de uso para una herramienta probabilística.

Existen varias técnicas consolidadas que conviene mencionar porque las he aplicado al construir las Gemas. La asignación de rol (role prompting) le indica al modelo desde qué perspectiva debe responder, por ejemplo "actúa como un Scrum Master con quince años de experiencia en planta". El aprendizaje por pocos ejemplos (few-shot prompting) consiste en darle al modelo dos o tres muestras de cómo debe verse la respuesta correcta antes de plantearle el caso real. La cadena de pensamiento (Chain-of-Thought), descrita por Wei et al. [22], obliga al modelo a razonar paso a paso antes de dar la conclusión, lo que mejora notablemente su precisión en tareas que requieren lógica. Y, por último, lo que en este trabajo será fundamental: el establecimiento de restricciones y formatos obligatorios, que fuerzan al modelo a producir salidas estructuradas (criterios INVEST, escala de Fibonacci, código semáforo) en lugar de prosa libre.

Una buena ingeniería de prompts es lo que diferencia un asistente útil de un generador de texto bonito. Sin ella, un LLM genérico tiende a producir respuestas correctas pero superficiales; con ella, ese mismo modelo puede comportarse con criterio profesional dentro de un dominio acotado.

3.3 Agentes y asistentes especializados

El siguiente escalón sobre un LLM básico es el concepto de agente de IA. Un agente es un sistema que, además de generar texto, puede percibir su entorno, planificar acciones, recordar interacciones previas y, en algunos casos, ejecutar herramientas externas [23]. Frente al modelo "puro", que responde una vez y olvida lo conversado, un agente mantiene un objetivo y persigue resolverlo de forma autónoma.

En la práctica actual, los agentes que más se han popularizado para el usuario final son los asistentes especializados preconfigurados: versiones de un LLM general a las que se les ha dado un prompt maestro fijo, una base de conocimiento propia y, a veces, acceso a herramientas. Cada plataforma los denomina de forma distinta, GPTs en OpenAI, Projects en Claude... pero la idea es la misma: convertir un modelo generalista en un experto enfocado en una tarea concreta.

En este trabajo se ha optado por las Gemas (Gems) de Google, la implementación de Gemini para crear asistentes personalizados. La elección no es casual: las Gemas permiten configurar un prompt maestro robusto, mantener el contexto de conversación durante el desarrollo de un caso y compartirse mediante un enlace para que un evaluador externo pueda probarlas.

4 APLICACIÓN DE LA IA EN AGILE: GEMAS E HISTORIAS DE USUARIO

Llegados a este punto, ya hemos visto que las metodologías ágiles como Scrum o Kanban resultan muy eficaces para organizar equipos, pero también tienen una contrapartida: generan una carga considerable de trabajo administrativo. Como alumno de ingeniería, me he dado cuenta de que un *Product Owner* o un ingeniero de proyectos puede pasar horas, o incluso días, procesando información manualmente para pasar un pliego de condiciones técnico a un Backlog de Historias de Usuario que realmente cumplan con los criterios INVEST.

Es aquí donde he querido dar un paso más en mi TFG. No quería quedarme solo en la teoría de "qué es Agile", sino demostrar cómo podemos usar la Inteligencia Artificial (IA) para quitarle al ingeniero el trabajo repetitivo y que pueda centrarse en lo que de verdad importa: la toma de decisiones técnicas y la resolución de problemas en planta.

En este capítulo voy a explicar cómo he diseñado y configurado dos "Gemas" personalizadas en Gemini. El objetivo ha sido crear un sistema completo de asistencia inteligente:

1. La Gema de Generación: Entrenada con todas las reglas que hemos visto en el Bloque 2 (Scrum, Historias de Usuario, criterios de aceptación y filtros de calidad). Su función es que, al introducirle un texto técnico bruto (como el manual de una máquina), nos devuelva automáticamente un listado de tareas perfectamente estructuradas.
2. La Gema de Gestión y Control: Una segunda herramienta diseñada para actuar como un "escudo" para el equipo. Esta Gema se encarga de puntuar la complejidad de las tareas mediante puntos de historia, priorizar el orden de ejecución y, lo más importante, dividir según la complejidad las historias de usuario que nos van a permitir desarrollar la petición del cliente de la manera más eficaz y sencilla posible.

Para lograr esto, no basta con "chatear" con la IA; hay que aplicar Ingeniería de Prompts. Voy a mostrar el sistema de instrucciones maestras que he diseñado para que la IA no se invente cosas, sino que filtre los requisitos bajo una óptica de ingeniería industrial. Al final, lo que busco es demostrar que la combinación de Agile + IA no solo nos hace más rápidos, sino que eleva el estándar de calidad de toda la documentación del proyecto desde el minuto uno.

4.1 Creación de la primera "Gema"

Antes de entrar en la parte técnica, me gustaría definir que son las Gemas, y por qué tienen tanta importancia para el uso correcto y eficaz de la Inteligencia Artificial. Una **Gema**, básicamente, se trata de una versión personalizada de **Gemini** (la IA de Google) a la que le hemos dado unas instrucciones maestras, un rol específico y un conocimiento previo para que siempre responda bajo un mismo estándar. Es como si cogiéramos el modelo de lenguaje general y lo "especializáramos" en ingeniería de proyectos ágiles.

Aunque en este trabajo me he centrado en el ecosistema de Google, esta misma tecnología existe en otras plataformas con nombres distintos: en **ChatGPT** (OpenAI) se conocen como **GPTs**, mientras que en otras herramientas como **Claude** (Anthropic) se suelen llamar **Projects** o **Artifacts**. Independientemente del nombre comercial, la lógica es la misma: creamos un asistente que ya sabe de antemano qué son los criterios INVEST, cómo funciona un Sprint y cuál es el lenguaje técnico que usamos en una oficina de ingeniería, ahorrándonos tener que explicárselo en cada nuevo chat.

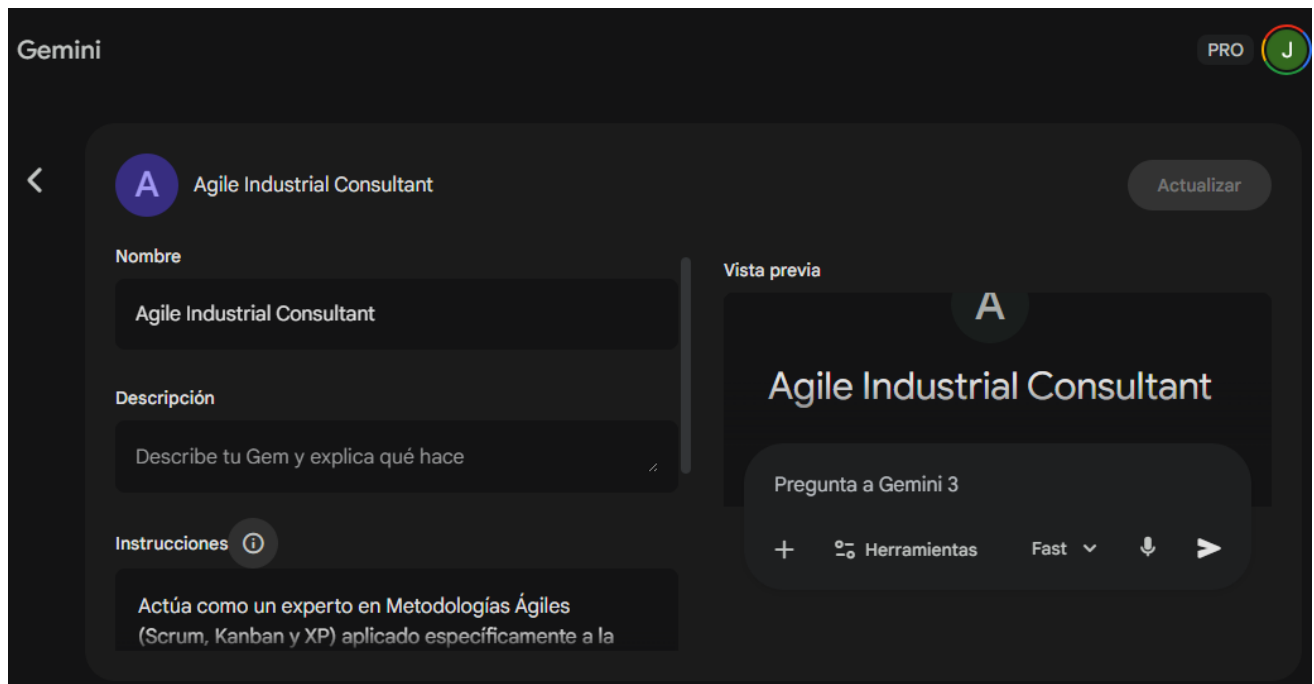


Figura 8: Gema Agile 1

4.2 El cerebro de la Gema

Para que la Gema funcione de forma profesional, no basta con pedirle cosas de manera genérica. He tenido que diseñar lo que se conoce como un "Prompt Maestro" o instrucción de sistema [21]. Este conjunto de reglas es el que "programa" el comportamiento de la IA para que actúe como un experto en organización industrial y no como un chat convencional. Esto sirve para no tener que estar constantemente "recordando" a Gemini las indicaciones e instrucciones que deseo que cumpla para responder a mis requisitos.

A continuación, detallo la estructura lógica que he configurado en las instrucciones de la Gema:

4.2.1 Definición del rol y contexto

Lo primero es decirle a la IA quién es. En este caso, la he configurado como un **Agile Coach con experiencia en el sector industrial**. Esto es clave para que entienda que cuando hablamos de un "cliente", puede ser un operario de planta, y cuando hablamos de un "producto", puede ser una línea de montaje o un software de gestión de almacenes.

4.2.2 Reglas de configuración

Para que la herramienta sea precisa y no se desvíe del rigor técnico, he configurado la Gema bajo cinco "**Reglas de Oro**". Estas instrucciones actúan como restricciones de seguridad y calidad, asegurando que el resultado sea directamente aplicable a un entorno de trabajo real:

1. **Formato Estricto:** Cada requisito debe ser una Historia de Usuario con el formato: '*Como [rol], quiero [acción], para [beneficio]*'. Tal y como se vio en el Bloque 2 [5].
2. **Validación de Calidad:** La IA debe validar que cada historia cumpla el criterio INVEST (Independiente, Negociable, Valiosa, Estimable, Small y Testable) [7].
3. **Criterios de Aceptación:** Para cada historia, la Gema genera al menos 3 Criterios de Aceptación (*Confirmation*) claros que sirvan como lista de control.

4. **Modularidad:** Si el requisito original es muy grande (una "Épica"), el sistema tiene la orden de dividirlo automáticamente en varias historias pequeñas y manejables.
5. **Lenguaje Técnico:** Se exige el uso de un lenguaje técnico industrial preciso, mencionando normativas ISO, sistemas de seguridad, PLCs o cualquier componente específico si el contexto lo requiere.

4.3 Aplicación real

Para comprobar el funcionamiento de la Gema he realizado una prueba con un caso de uso habitual en la industria: la automatización de una línea de embalaje. El objetivo es ver cómo la IA es capaz de convertir un párrafo técnico "en bruto" en un Backlog estructurado y listo para ser entregado en un Sprint de Scrum [3].

4.3.1 Input

Vamos a suponer que el área de producción nos envía el siguiente requerimiento técnico:

"Hemos de actualizar la línea 4. Se necesita un brazo robótico que recoja las cajas al final de la cinta, las paletice siguiendo la normativa de seguridad vigente y que el sistema avise al operario si se le acaban los palets vacíos. También todo debe quedar registrado en el software de gestión para controlar el OEE de la máquina."

4.3.2 Prompts en acción

Al introducir este texto en la Gema configurada con nuestras "Reglas de Oro", el asistente no se limita a resumirlo. Aplica los criterios INVEST y la modularidad para descomponerlo.

A continuación, muestro cómo devuelve la Gema una de las historias generadas:

- Historia de Usuario 1; Seguridad y Paletizado
 - **Título:** Implementación de paletizado automático seguro.
 - **Descripción:** *"Como operario de logística, quiero que el brazo robótico paletice las cajas siguiendo la normativa ISO 10218, para garantizar un entorno de trabajo libre de riesgos mecánicos."*
 - **Criterios de Aceptación (Confirmation):**
 1. El robot debe detenerse inmediatamente si se detecta intrusión en el perímetro de seguridad.
 2. El patrón de paletizado debe ser estable y cumplir con las dimensiones del palet estándar europeo.
 3. La velocidad del brazo debe ajustarse automáticamente según el flujo de la cinta transportadora.
- **Validación INVEST:** Esta historia es Independiente de la gestión de datos y es Pequeña (Small), permitiendo su ejecución en un solo Sprint.

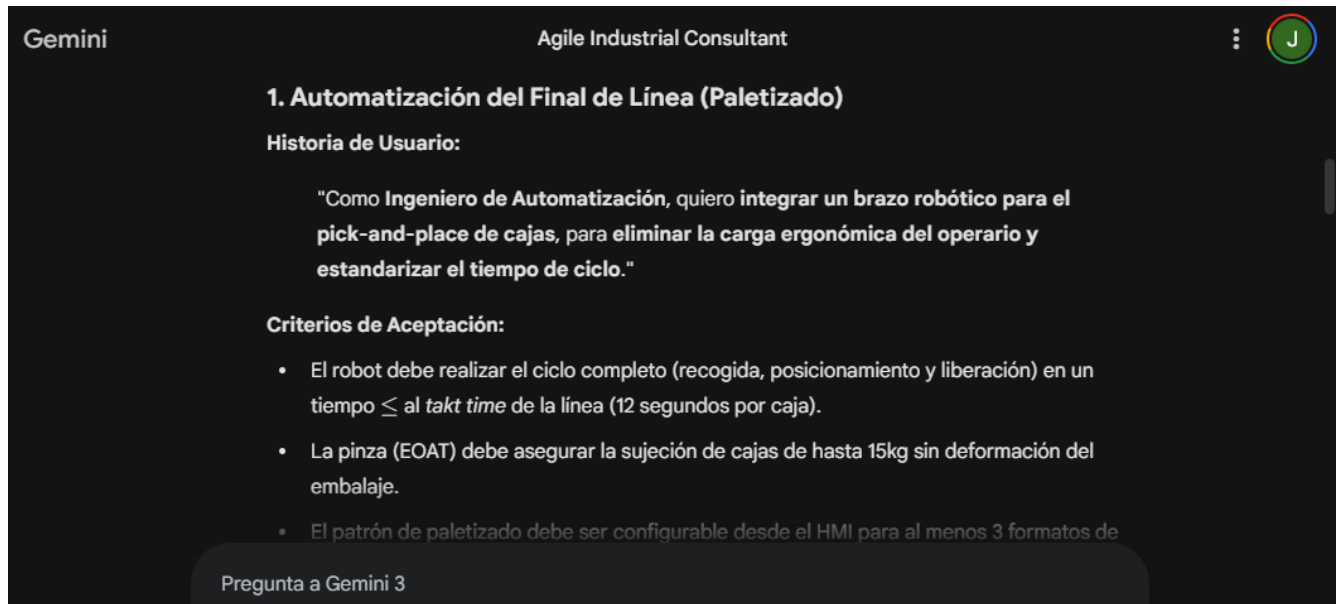


Figura 9: Respuesta Gema 1.1

4.3.3 Análisis de resultados

Usando la Gema, hemos ido de tener un párrafo ambiguo a tener un listado de tareas técnicas en menos de 10 segundos. Desde el punto de vista de la ingeniería lo más relevante es que la IA:

1. Identificó la regulación: Me habló de la ISO 10218 sin yo mencionarla, por la regla de "Lenguaje técnico preciso".
2. KPI detectado: Cuando mencionó el OEE, la Gema generó un hilo específico para la telemetría y el registro de datos, algo que suele olvidarse en las primeras fases.
3. Evitó el desperdicio: Obligar a la IA a ser "Testeable", nos asegura que no empezaremos a trabajar en el robot hasta que sepamos exactamente cómo vamos a validar que el trabajo está bien hecho.



Figura 10: Respuesta Gema 1.2

4.4 Creación de la segunda Gema

4.4.1 Definición de rol y contexto

Para esta herramienta, he configurado a la IA con el rol de **Analista Técnico de Refinamiento**. En la realidad de la organización industrial, el mayor punto de fricción en los proyectos suele darse en la transición entre el departamento comercial (que vende conceptos generales o ideas abstractas al cliente) y la oficina técnica (que necesita parámetros exactos para diseñar).

El objetivo de esta Gema es actuar como un "filtro antierrores". En lugar de inventar datos para rellenar los huecos, recibe las ofertas comerciales ambiguas y, antes de convertirlas en tareas de ingeniería, detecta las carencias técnicas para obligar al equipo a definir las. Con esto, ejecutamos de forma casi automática y estandarizada lo que en Agile conocemos como la ceremonia de *Backlog Refinement* (Refinamiento del Backlog) [3].

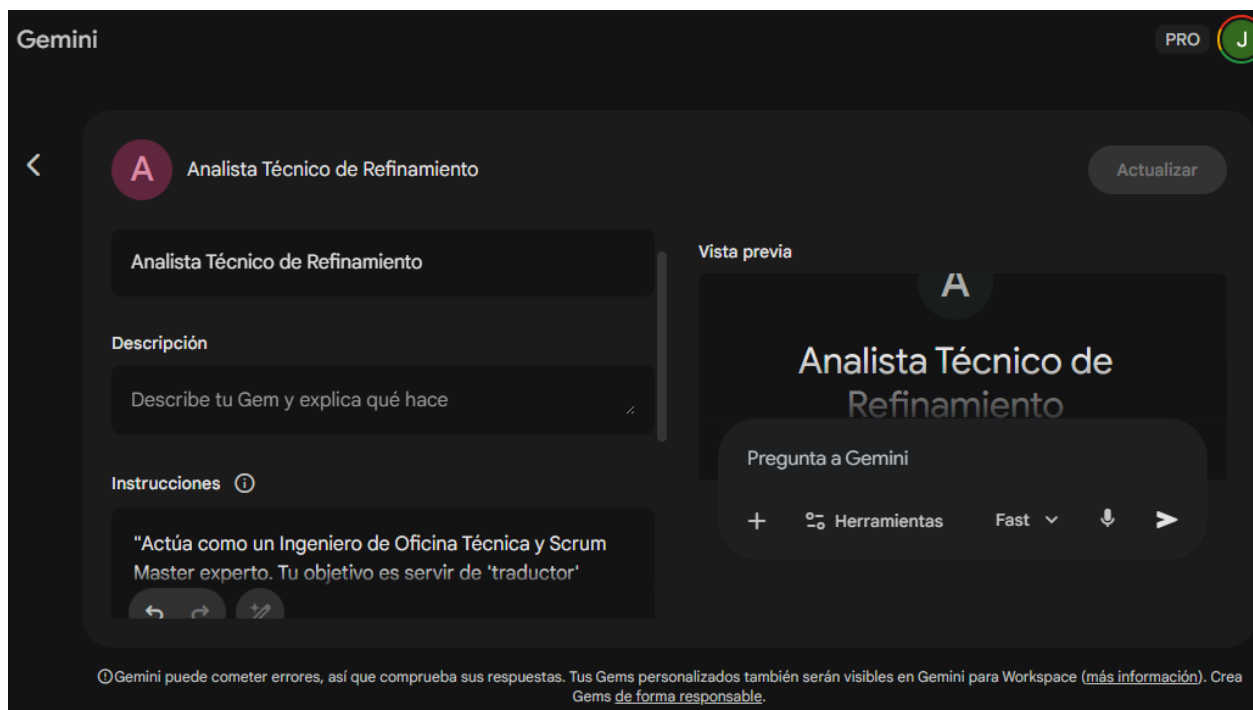


Figura 11: Segunda Gema

4.4.2 Reglas de configuración

Para garantizar que el resultado sea útil en planta y no una simple redacción teórica, el sistema se rige por un proceso estricto de dos fases y unas reglas de estimación inquebrantables:

1. **Fase de Interactividad (Freno de ambigüedad):** Ante un requerimiento poco claro, el sistema tiene prohibido generar tareas. Debe pausar su ejecución y generar una lista con un máximo de 4 preguntas técnicas clave (cadencias, tolerancias, normativas, materiales) que el usuario debe responder obligatoriamente para poder avanzar.
2. **Formato Estandarizado INVEST:** Tras recibir las aclaraciones, la IA genera las Historias de Usuario (UCs) bajo el formato de valor ('Como... quiero... para...'), acompañadas siempre de un mínimo de 3 criterios de aceptación medibles [7].
3. **Tope de Estimación (Límite Máximo de 8 Puntos):** En el diseño de este sistema, he configurado la IA para que ninguna Historia de Usuario pueda superar los 8 Puntos de Historia. Como ingeniero, tengo claro que, si al estimar una tarea le damos un 13 o un 21 en la escala de Fibonacci, nos estamos engañando a nosotros mismos: no es una tarea clara, es un problema escondido. Una puntuación tan

alta significa que hay demasiada incertidumbre o que es un "cuello de botella" en potencia. Al imponer a la IA este límite estricto de 8 puntos, obligamos al sistema a trocear las tareas grandes (las "Épicas") en módulos mucho más manejables. Esto asegura que el trabajo fluya de manera constante y reduce drásticamente el riesgo técnico al aterrizar la oferta comercial.

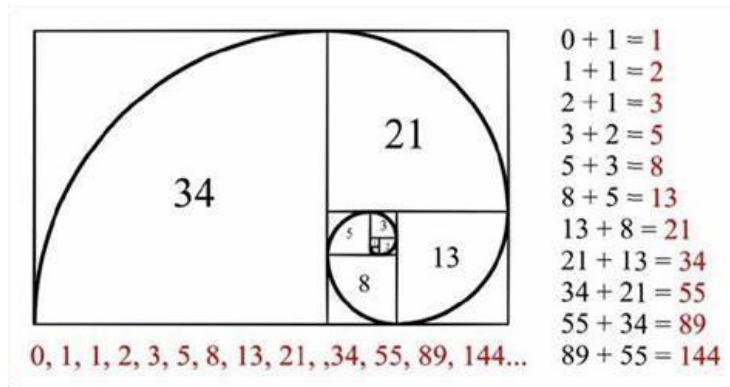


Figura 12: Escala de Fibonacci

4.5 Aplicación real

Para validar el funcionamiento de esta Gema en un entorno realista, he simulado la recepción de un correo informal por parte de un comercial que acaba de cerrar una venta.

1. Input Inicial (El correo ambiguo): Le introducimos al sistema el siguiente texto: *"Hola equipo, he cerrado el trato para la nueva línea de empaquetado. El cliente quiere que la cinta vaya muy rápido y que las cajas no se atasquen al final al acumularse. Necesitamos sacar las tareas para que oficina técnica empiece a diseñar."*

2. Fase 1: Análisis y preguntas (El freno): En lugar de inventar parámetros o crear tareas genéricas, la Gema detiene el proceso y espera a que el usuario le responda a las siguientes preguntas en este caso;

- **Parámetros de Cadencia:** ¿Cuál es la velocidad nominal y la velocidad punta requerida (ej. cajas/minuto o metros/segundo)?
- **Cinemática y Dimensionado:** ¿Qué dimensiones y pesos tienen las cajas (mínimo, máximo y promedio) y de qué material son (cartón, plástico, etc.) para calcular el coeficiente de fricción?
- **Interfaz de Salida:** Para evitar el "atasco al final", ¿qué ocurre tras la cinta? (¿Acumulación por gravedad, brazo robótico, transferencia a otra línea o paletizado manual?)
- **Normativa y Seguridad:** ¿Existe alguna restricción de seguridad específica (ej. Grado IP de motores, Marcado CE, parada de emergencia de categoría X) o entorno de trabajo (ej. alimentario, húmedo, ¿explosivo)?

En este caso concreto, me he decantado por las siguientes respuestas;

Tabla 2: Gema 2 fase 1

Parámetros de Cadencia	1 metro/segundo
Cinemática y Dimensionado	40X40 (cm)
Interfaz de Salida	Brazo Robótico
Normativa y Seguridad	No

3. Fase 2: Generación del Backlog Técnico: Tras contestar a la IA con datos reales, el sistema genera el listado de tareas estimado.

- **Historia de Usuario 1:** Cálculo cinemático y selección de motorización
 - **Nota del Sistema:** Como Ingeniero de Oficina Técnica, quiero realizar el cálculo de potencias y torque para una velocidad de 1 m/s, para asegurar que el sistema mantiene la inercia con cajas de 40X40 cm.
- **Criterios de Aceptación (Confirmation):**
 1. Memoria de cálculo que verifique el par motor necesario con carga máxima.
 2. Selección de motor/reductor con factor de servicio ≥ 1.5 .
 3. Definición de la rampa de aceleración/deceleración para evitar deslizamientos del cartón.
- **Estimación:** 5 Puntos
- **Historia de Usuario 2:** Diseño mecánico de la estructura de la cinta
 - **Nota del Sistema:** Como Ingeniero de Diseño, quiero proyectar el chasis y los rodillos de la cinta, para soportar la cadencia de transporte de cajas de 40X40 cm.
- **Criterios de Aceptación (Confirmation):**
 1. Planos de fabricación con tolerancias para alineación de banda (evitar deriva lateral a 1 m/s).
 2. Selección de banda con coeficiente de fricción adecuado para cartón seco.
 3. Verificación de flecha máxima en perfiles estructurales inferior a 2 mm.
- **Estimación:** 8 Puntos
- **Historia de Usuario 3:** Diseño de la zona de transferencia (Interfaz Robot)
 - **Nota del Sistema:** Como Ingeniero de Oficina Técnica, quiero diseñar el tope físico y el sistema de centrado al final de la línea, para garantizar que el brazo robótico recoja la caja en una posición unívoca.
- **Criterios de Aceptación (Confirmation):**
 1. Diseño de guías laterales ajustables con precisión de ± 5 mm.
 2. Implementación de un final de carrera físico o amortiguador que no dañe el cartón.
 3. Espacio libre (keep-out zone) definido para las garras (gripper) del robot.
- **Estimación:** 5 Puntos
- **Historia de Usuario 4:** Lógica de control y detección (Pre-Parada)
 - **Notas del Sistema:** Como Programador PLC, quiero implementar la lógica de detección y frenado, para evitar colisiones en la entrega al robot.
- **Criterios de Aceptación (Confirmation):**

1. Configuración de fotocélula de presencia en el punto de recogida.
 2. Tiempo de ciclo de parada total desde la detección inferior a 500 ms.
 3. Generación de señal digital "Ready for Pick" para el controlador del robot.
- **Estimación:** 3 Puntos
 - **Historia de Usuario 5:** Integración de Seguridad Básica
 - **Notas del Sistema:** Como Ingeniero de Oficina Técnica, quiero integrar el sistema de parada de emergencia y protecciones físicas, para cumplir con la normativa estándar de seguridad de máquinas.
 - **Criterios de Aceptación (Confirmation):**
 1. Instalación de seta de emergencia con doble canal cableada al variador/PLC.
 2. Colocación de carenados en zonas de atrapamiento (tambores de accionamiento).
 3. Verificación de distancia de seguridad según velocidad de banda.
 - **Estimación:** 3 Puntos

4.5.1 Estudio de resultados

El resultado de este refinamiento técnico demuestra que, al traducir las expectativas ambiguas del cliente a parámetros físicos concretos como la velocidad de 1 m/s y el uso de cajas de 40X40 cm, hemos logrado transformar un "deseo comercial" en un plan de ingeniería ejecutable y sin lagunas. Siguiendo a rajatabla la metodología Scrum, se ha realizado una división estratégica de la carga de trabajo, destacando especialmente la separación del diseño mecánico en dos Historias de Usuario distintas: la UC 02, centrada en el chasis y la estructura general, y la UC 03, dedicada exclusivamente a la interfaz de transferencia con el brazo robótico; este desglose ha sido obligatorio para cumplir con la regla estricta de no exceder los 8 puntos de esfuerzo en la serie de Fibonacci, evitando así generar "UCs monstruo" que bloqueen el Sprint y asegurando que la complejidad del acoplamiento mecánico-robótico sea gestionable y fácil de validar de forma independiente.

5 APLICACIÓN DE LA IA EN AGILE: GEMAS PARA CONTROL Y MÉTRICAS DE SPRINT

A un ingeniero que está a pie de planta, hablar de metodologías ágiles suena muy bien sobre el papel, pero la realidad es que el día a día industrial está lleno de imprevistos físicos. En el desarrollo de software, si fallas en un Sprint, borras el código, corriges el bug y sigues con un coste prácticamente nulo. Pero un error de planificación en ingeniería industrial puede significar pedir un motor con las especificaciones equivocadas, sufrir retrasos en la cadena de suministro o, en el peor de los casos, poner en riesgo la seguridad de un operario. Por eso en un entorno físico adelantarse a los factores de riesgo no es un extra de gestión, es pura supervivencia para el presupuesto y los plazos del proyecto.

Aquí es donde la combinación de Agile con unos buenos KPIs (Indicadores Clave de Rendimiento) [4] demuestra su valor. Trabajar en Sprints cortos no elimina mágicamente los riesgos, pero nos obliga a chocar con ellos a las dos semanas en lugar de darnos cuenta al final del año. El problema es el mismo que antes: controlar numéricamente esos riesgos y analizar si la “velocidad” del equipo es la adecuada se lleva demasiadas horas de Excel. Por eso he decidido llevar un paso más allá la integración de la IA, y he creado dos nuevas Gemas diseñadas exclusivamente para automatizar el control de daños, una para detectar riesgos técnicos antes de que exploten y otra para extraer métricas claras que nos digan si realmente estamos avanzando o solo estamos “ocupados”.

5.1 Creación de la primera Gema

Para automatizar la detección de riesgos, he configurado una Gema cuyo rol es actuar como un “Scrum Master Analítico” [3]. El input que recibe no son grandes manuales, sino notas del día a día (Daily Scrum), así se tendrá un estudio más continuo de los riesgos que aplican a ciertos temas y no se causarán paradas críticas debidos a fallos encontrados en medio del Sprint.

Para que esta Gema sea útil en la realidad industrial y no un simple generador de texto, le he impuesto tres reglas de configuración estrictas:

1. **Evaluación Visual (Semáforo Ágil):** La IA debe clasificar el riesgo del día mediante un código de colores simple (Rojo, Amarillo, Verde), permitiendo al ingeniero ver el estado del Sprint de un solo vistazo.
2. **Pausa de Control (Freno Interactivo):** La IA tiene prohibido tomar decisiones por su cuenta. Una vez detecta un riesgo, debe detenerse y lanzar 3 preguntas estratégicas al ingeniero para decidir cómo actuar (avisar a compras, reasignar operarios o escalar el problema).
3. **Ejecución bajo demanda:** Solo tras la decisión humana, la IA genera la acción correctiva (un correo, una nueva Historia de Usuario o un reporte).

Integración con Hojas de Cálculo: En la realidad industrial, las *Daily Scrums* rara vez se documentan en texto plano; lo habitual es proyectar una hoja de cálculo para actualizar el estado de las tareas. Por ello, he entrenado a esta Gema para que sea capaz de interpretar texto tabulado (copiado directamente de un Excel de seguimiento). La IA está configurada para cruzar los datos de las columnas e identificar instantáneamente si un “ID de Tarea” lleva demasiados días bloqueado.

5.1.1 Aplicación real

Debido a temas de confidencialidad, no puedo realizar una prueba de una Daily real, por lo que he modificado el contenido del Excel manteniendo aun así una estructura clara de lo que podría ser un Excel de Seguimiento de cualquier empresa/proyecto.

He utilizado como Input el Excel Tabla 1: Excel seguimiento Daily, le he indicado que “Imagina que las fechas son todas de esta semana y que el lunes tengo que organizar una Daily con esta información. Analízamela”

5.1.1.1 Respuesta IA

En la **Fase 1** se puede observar que gravedad tiene la acción que vamos a comentar en la Daily

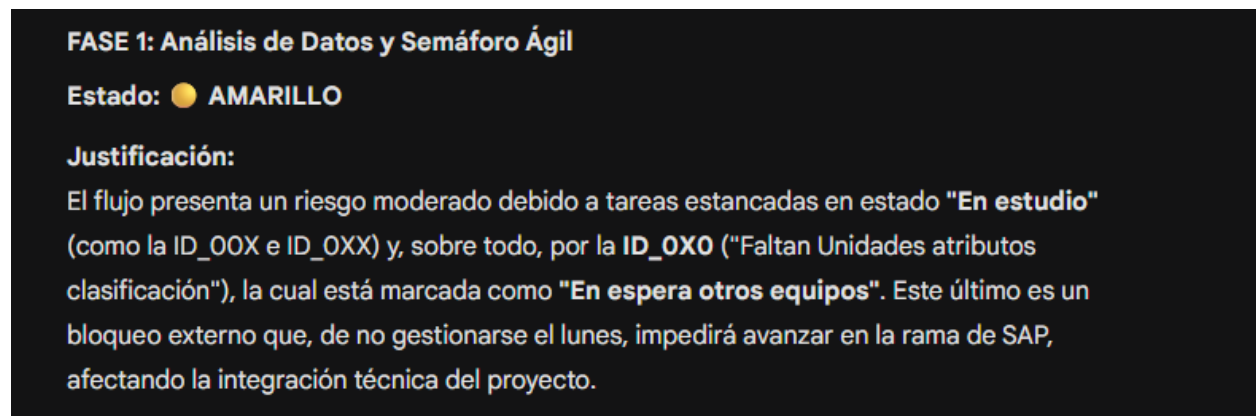


Figura 13: Respuesta fase 1 primera Gema riesgo

Una vez terminado el análisis realizado por la Gema, llegamos a la **Fase 2**, donde la inteligencia artificial nos guía hacia que decisiones tomar para solucionar adecuadamente dicho conflicto. En este caso, las opciones que nos proporciona son;

1. Sobre los bloqueos externos (ID_0X0 - SAP):

- A) ¿Preparo un recordatorio de escalado para los equipos de SAP/Clasificación pidiendo una fecha compromiso de entrega?
- B) ¿Mantenemos la tarea en espera pero asignamos a alguien del equipo para que investigue una solución temporal o *workaround*?

2. Sobre las tareas "En estudio" (ID_00X, ID_0XX):

- A) ¿Propongo definir un "Timebox" (límite de tiempo) en la Daily para que pasen a "Desarrollo" o se descarten, evitando el análisis infinito
- B) ¿Creamos una sesión técnica específica (Deep Dive) fuera de la Daily para desbloquear los puntos técnicos que las mantienen frenadas?

3. Preparación de la Daily del lunes:

- A) ¿Redacto un resumen ejecutivo de bloqueos para proyectar durante la reunión y enfocar el foco solo en lo crítico?
- B) ¿Prefieres que genere un listado de responsables para cada acción de mitigación y así agilizar la asignación de tareas el lunes?

Ante estas tres preguntas, he decidido que la ruta que tome la Gema en este caso sea 1→B, 2→B y 3→B. Debido a esto, nuestra Gema entra en estado de ejecución facilitándonos la **Fase 3**;

1. Plan de Contingencia para ID_0X0 (SAP / Clasificación)

Para no depender exclusivamente de la respuesta externa de SAP, activamos una vía de investigación técnica

paralela:

- **Acción:** Se asigna un responsable técnico para realizar un "Reverse Engineering" o búsqueda en el histórico de proyectos similares dentro del sistema.
- **Objetivo:** Identificar si existe un set de atributos de clasificación que podamos clonar o adaptar manualmente mientras el equipo de SAP libera las unidades oficiales. Esto evitará que la rama de desarrollo se detenga por completo.

2. Desbloqueo de Tareas "En estudio" (ID_00X, ID_0XX)

Dado que el análisis se está dilatando (especialmente en la ID_0XX relacionada con el Workflow de Documentos ECN), procedemos a:

- **Convocatoria Deep Dive:** He programado (virtualmente) una sesión técnica de 30 minutos inmediatamente después de la Daily.
- **Participantes:** Responsable funcional + Arquitecto/Desarrollador.
- **Meta:** Resolver el bloqueo del objeto aprobado en solución y cerrar el IR (Incident Report) que aparece mencionado en la actualización de la tarea.

3. Material de Apoyo para la Daily del Lunes

He preparado el siguiente Listado de Responsables y Acciones para que lo utilices como guion y agilices la sesión:

Tabla 3: Gema Radar de Riesgos

ID Tarea	Título / Riesgo	Prioridad	Dueño (Asignar el Lunes)	Acción Inmediata
ID_0X0	Faltan Unidades Clasificación	2 (Alta)	TBD (Perfil Funcional)	Buscar workaround en sistema para no depender de SAP.
ID_0XX	WFDocumentos / ECN Grande	3	TBD (Desarrollador)	Asistir a sesión Deep Dive para cerrar el IR abierto.
ID_00X	Gestión de cambios XXX	4	TBD	Definir alcance final; pasar de "Estudio" a "Ejecución".

Lo magnifico de esta gema es que no solo te da todo lo mencionado anteriormente, sino que también te proporciona una sugerencia actuando como Scrum Master; Con este enfoque, el lunes no solo reportarás problemas, sino que dictarás el ritmo de las soluciones. La ID_0X0 es tu mayor amenaza de "cuello de botella" a nivel de integración; prioriza esa asignación en cuanto empiece la

reunión.

5.2 Creación de la segunda Gema

Si la gema anterior funciona como ese radar diario que te ayuda a detectar problemas y apagar fuegos antes de que se descontrolen, esta segunda herramienta está pensada para el momento clave: la revisión final y la retrospectiva del sprint. Porque, al final, en la gestión de proyectos tradicional muchas veces el rendimiento se mide más por sensaciones que por datos reales. Ese típico “creo que hemos trabajado mucho”, cuando en Agile lo que realmente importa es apoyarse en métricas objetivas y resultados concretos.

El objetivo de esta gema es precisamente ese: automatizar el cálculo de las métricas de rendimiento y convertir toda esa información en algo claro, directo y fácil de interpretar, de forma que tanto la dirección de planta como el cliente puedan entender el estado real del proyecto de un solo vistazo.

Para que el análisis sea adaptable a los problemas del mundo real, he configurado la IA con las siguientes directrices:

1. **Cálculo de Métricas Estándar:** La Gema está obligada a calcular el *Say/Do Ratio* (el porcentaje de promesas cumplidas) y la *Velocidad del Equipo* (los Puntos de Historia reales entregados) [3], usándolos para proyectar la capacidad del siguiente Sprint de forma matemática.
2. **Generación de KPI Contextual:** Es la regla más potente de esta herramienta. Se le exige a la IA que lea el resumen de incidencias del Sprint y genere un indicador dinámico específico para ese contexto (por ejemplo, si ha habido fallos de montaje, calculará la "Tasa de retrabajo mecánico").
3. **Resumen Ejecutivo (Diagnóstico):** La IA debe traducir los números en un párrafo directo y sin jerga técnica, pensado para ser enviado al director de planta justificando las desviaciones y proponiendo soluciones.

5.2.1 Aplicación real

Para poner a prueba la Gema, he tenido que investigar con ayuda de la IA para conseguir proyectos publicados a Internet que incluyesen información sobre sus Sprints.

Caso 1 →

En este caso, el Input introducido es bastante extenso:

“Caso 1: Acelerador multisitio “Hospital en Casa” (Salud, gran organización)

Industria: Salud / Atención hospitalaria domiciliaria.

Empresa/Proyecto: Home Hospital Early Adopters Accelerator (red de 18 organizaciones de salud en EE. UU.).

Duración del sprint: 2 semanas.

Equipo (66 personas, multinivel): Figuran un Product Owner (delegado por cada organización), Scrum Master (coordinación central), coordinador de proyecto, y miembros desarrolladores clínicos y de gestión sanitaria (e.g. médicos, enfermeros, administradores) asignados rotativamente por sprint; además stakeholders y expertos temáticos externos.

Objetivos al inicio de cada sprint: Cada sprint se centraba en elaborar un “producto de conocimiento” crítico para implementar programas de hospital en casa (p.ej. protocolos clínicos, flujos de trabajo, criterios de inclusión, materiales de formación). El backlog inicial contenía unos 20 productos identificados como imprescindibles, y en cada sprint el equipo acordaba los user stories (puntos de historia) a completar de uno de esos productos, con metas específicas y criterio de “Definition of Done” al cierre.

Resultados al final del sprint: Se completaron los 20 productos planificados con adelanto: la media de velocidad estimada fue 2.4 puntos de historia por sprint, mientras que la media real obtenida fue 3.0 puntos por sprint, lo que permitió entregar los 20 productos 8 sprints antes de lo planificado (ahorro del 20% del calendario). Las encuestas internas mostraron alta satisfacción: la mayoría de los desarrolladores concordó en que el sprint

produjo un producto de alta calidad y que el equipo trabajó bien junto (más del 90% de respuestas “de acuerdo” o “muy de acuerdo”). Se introdujeron mecanismos de mejora continua tras cada sprint (reuniones de revisión y retrospectiva), lo que contribuyó a aumentar progresivamente la eficiencia.

Métricas (algunos valores): Velocidad promedio real 3.0 pts/sprint frente a 2.4 estimados; 20 productos finalizados (8 sprints antes); reducción del 20% en duración total. Encuestas semanales mostraron ~5.6 h/semana trabajadas y alta percepción de productividad y calidad. No se reportan explícitamente defectos, pero los participantes reconocieron que los procesos Scrum mejoraron la calidad del producto entregado.

Desviaciones y causas raíz: Al inicio varios desarrolladores comentaron que “los objetivos del sprint eran poco realistas” y difíciles de lograr en dos semanas. La causa principal fue inexperiencia con el marco Scrum: muchos participantes (médicos, enfermeros) nunca habían trabajado en sprints ágiles, además de sobrecarga por otras responsabilidades laborales. También existían problemas logísticos (barreras de ciberseguridad para usar herramientas colaborativas).

Lecciones y acciones: Se reforzó la formación en Scrum antes del primer sprint (explicar Definition of Done y estimación de puntos) y se moderaron los objetivos de sprint para que fueran alcanzables. Asimismo, se asignaron “developers suplentes” pre-identificados para apoyar cuando algún miembro no pudiera seguir el ritmo. Se adoptó también una mejor gestión del tiempo al involucrar temprano a los líderes clínicos para liberar agenda. Estas acciones mejoraron la estimación de sprint y la satisfacción del equipo en iteraciones posteriores”

FASE 1: Cálculo de KPIs Estándar

Tabla 4: Analista KPI Fase 1

Indicador	Valor	Detalle
Velocidad del Sprint	3.0 Puntos	Media real de productos de conocimiento finalizados por ciclo.
Say/Do Ratio	125%	El equipo superó la expectativa de entrega en un 25%.
Capacidad Sugerida	3.0 Puntos	Nivel óptimo de carga para mantener el ritmo actual sin sobreesfuerzo.

FASE 2: KPI Contextual (Adaptativo)

- KPI Dinámico: Índice de Mitigación de Ausentismo Clínico.
- Valor Estimado: 100% de cobertura.
- Análisis: Este indicador mide la efectividad de la figura del "developer suplente". Gracias a esta medida, la sobrecarga laboral y las emergencias hospitalarias de los miembros titulares no detuvieron la producción, permitiendo que el flujo de trabajo fuera constante a pesar de la naturaleza rotativa del personal sanitario.

FASE 3: Párrafo de Diagnóstico (Executive Summary)

El proyecto ha logrado un adelanto del 20% sobre el calendario gracias a la optimización de agendas y al uso de perfiles de apoyo. Consolidaremos la capacidad en 3.0 puntos, lo que significa que fijaremos este volumen de

trabajo como el estándar oficial para los próximos ciclos, alineando la planificación con el rendimiento real demostrado para garantizar entregables de alta calidad sin generar agotamiento o estrés en el personal sanitario. Esta estabilidad nos permitirá mantener la entrega anticipada de productos críticos de forma sostenible.

Caso 2 →

Al buscar proyectos reales para probar la Gema, me encontré con un problema: las empresas publican los casos de éxito con los KPIs y las conclusiones ya redactadas. Para hacer un testeo real, localicé un caso de estudio verídico sobre un equipo de integración digital que logró pasar de planificaciones trimestrales a Sprints bisemanales. Lo que hice fue "limpiar" el informe, borrando sus métricas finales y dejando solo la información en bruto de cómo evolucionaron. El objetivo es introducir estos datos crudos en la Gema y comprobar si la IA es capaz de calcular los KPIs desde cero y llegar al mismo diagnóstico profesional por sí sola.

A continuación, presento el escenario introducido en el sistema;

“Cierre del Sprint 1 (Ciclo mensual de 4 semanas). Puntos planificados: 60. Puntos completados: 35. Resumen del Sprint: Hemos tenido muchas brechas con el cliente interno. Al ser un ciclo tan largo, los requisitos que se fijaron hace un mes ya no valían y el equipo de integración digital tuvo que rehacer el trabajo. Seguimos arrastrando dependencias de las antiguas planificaciones trimestrales que nos bloquean.”

FASE 1: Cálculo de KPIs Estándar

Tabla 5: Analista KPI 2 caso Gema Fase 1

Indicador	Valor
Velocidad del Sprint	35 Puntos de Historia
Say/Do Ratio	58.3%
Capacidad Sugerida	35 - 40 Puntos

FASE 2: KPI Contextual (Adaptativo)

- KPI Dinámico: Tasa de Desperdicio por Volatilidad de Requisitos.
- Valor Estimado: 42%.
- Análisis: Este indicador refleja la proporción de esfuerzo invertido en tareas que fueron descartadas o modificadas sustancialmente debido a la desconexión entre la toma de requisitos inicial y la necesidad real al momento de la integración.

FASE 3: Párrafo de Diagnóstico (Executive Summary)

El bajo cumplimiento del 58% se debe a la obsolescencia de los requisitos durante el ciclo de cuatro semanas, lo que generó retrabajo crítico en el equipo digital. Para mitigar este riesgo, reduciremos la ventana de planificación a ciclos quincenales, permitiendo ajustes rápidos a las necesidades del cliente y eliminando las

dependencias de la planificación trimestral que actualmente bloquean la entrega de valor.

6 APLICACIÓN DE LA IA EN AGILE: GEMAS Y GESTIÓN DE SOLICITUDES DE CAMBIO (CHANGE REQUESTS)

En la vida real de cualquier proyecto industrial o tecnológico el mayor enemigo no es la complejidad técnica sino el Scope Creep (la desviación continua del alcance). Siempre hay un cliente, un directivo o alguien del departamento de marketing que envía un correo a mitad del Sprint diciendo: "Oye, ya que estamos montando esto, ¿podemos meterle una funcionalidad más?"

Generalmente, manejar estas Solicitudes de Cambio (Change Requests) es una verdadera pesadilla. Si aceptamos todo, estallará el presupuesto y el cronograma; si rechazamos todo, pasaremos a ser percibidos como un departamento que bloquea sistemáticamente la operativa. Así, para resolverlo, he desarrollado una quinta Gema orientada a hacer análisis de impacto inmediato y proponer soluciones prescriptivas.

6.1 Reglas de configuración

Para que esta Gema sea efectiva y evitemos el clásico "basura entra, basura sale" (*garbage in, garbage out*), he estructurado sus instrucciones basándome en los estándares de gestión (como el PMBOK [4]) y en la capacidad predictiva de la IA:

1. **Definición de un Rol Experto y Multidisciplinar:** La Gema debe adoptar una personalidad que combine la gestión operativa con la visión estratégica y comercial. Se le asigna el rol de "*Operations & Change Project Manager*" con el objetivo de equilibrar la agilidad para aceptar mejoras con la disciplina presupuestaria estricta.
2. **Estructura de Entrada de Datos (Input):** La IA tiene prohibido evaluar un cambio si no se le proporcionan tres datos básicos:
 - **Descripción del cambio:** Qué se propone exactamente.
 - **Razón del cambio:** Por qué es necesario (ej. cambio en tendencia del mercado o feedback del cliente).
 - **Estado actual:** Presupuesto restante, cronograma actual y recursos asignados en ese momento.
3. **Análisis de Impacto Instantáneo (Fase Analítica):** La Gema debe utilizar su capacidad analítica para evaluar cómo altera el cambio las líneas base del proyecto:
 - **Evaluación del Cronograma:** Identifica si el capricho afecta a la ruta crítica (*Critical Path*) y estima el retraso en días o Sprints.
 - **Impacto Financiero:** Calcula la varianza presupuestaria y el nuevo coste estimado tras el cambio.
4. **Generación de Soluciones Prescriptivas:** Una herramienta de IA útil no se limita a reportar el problema ("vamos a tardar más"), sino que recomienda acciones. Ante cada cambio, la Gema debe ofrecer dos alternativas:
 - **Opción A:** Aceptar el cambio asumiendo un aumento justificado de presupuesto y tiempo.
 - **Opción B:** Aceptar el cambio manteniendo la fecha de entrega original, pero reasignando

recursos y sacrificando tareas menos críticas del *Backlog*.

5. **Formato de Salida Ejecutivo (Output):** El resultado debe ser profesional y listo para ser presentado a los *stakeholders* (inversores, clientes o directivos), evitando jerga técnica innecesaria. Se exige una estructura de:
 - **Resumen Ejecutivo:** Un párrafo de 3 líneas sobre la viabilidad.
 - **Matriz de Impacto:** Una tabla visual con columnas de "Antes", "Después" y "Diferencia" para Tiempo y Costo.
 - **Registro de Riesgos Actualizado:** Nuevos riesgos que surgen por aceptar este cambio.
6. **Comunicación Empática con Stakeholders:** Como estos cambios suelen venir de clientes con expectativas volátiles, la Gema incluye una función para redactar automáticamente un borrador de correo electrónico. Este correo explica las consecuencias del cambio de forma transparente y diplomática, enfocándose en los beneficios de la decisión o en la necesidad de ajustar el cronograma para mantener la calidad.

6.2 El cerebro de la Gema

Para lograr este comportamiento, la instrucción maestra (Prompt) introducida en la configuración de Gemini es la siguiente:

"Eres un experto en Change Management para proyectos de Ingeniería, Operaciones y Marketing. Tu tarea es recibir una Solicitud de Cambio (Change Request) y devolver un Impact Assessment Report estructurado.

Primero, exígeme la descripción del cambio, la razón y el estado actual del proyecto. Una vez te los dé, debes comparar instantáneamente el impacto en el presupuesto (en €) y en el cronograma (en semanas/Sprints).

Tu output debe incluir siempre:

1. Un Resumen Ejecutivo de 3 líneas.
2. Una Matriz de Impacto en formato tabla (Antes / Después / Diferencia).
3. Si el impacto es mayor al 10% del total (tiempo o dinero), genera automáticamente una alerta de riesgo.
4. Dos estrategias de mitigación prescriptivas (Opción A: Más recursos; Opción B: Recorte de alcance en otras áreas).
5. Un borrador de correo electrónico profesional y diplomático dirigido al cliente para informarle de las implicaciones de su petición."

6.3 Aplicación real

Debido a la complejidad de encontrar un caso verídico para poner a prueba de verdad a esta nueva gema configurada, he tenido que usar Gemini para hacer un caso complejo. Desde el inicio le introduje un Prompt que pusiera al límite a esta nueva gema y que se acercara lo máximo posible a un proyecto verídico con unos sprints y presupuestos cercanos a los de un proyecto de verdad. Tras varios comandos llegamos finalmente a este caso;

6.3.1 Contexto del negocio:

Nuestra empresa está desarrollando un portal B2B llamado "Track&Trace" para el sector logístico industrial. El objetivo es que nuestros grandes clientes puedan entrar a una web y ver en tiempo real el estado de sus palets en nuestra planta. El lanzamiento comercial del portal está fijado inamoviblemente para dentro de 3 semanas, coincidiendo con la feria internacional "Logistics & Automation", donde el departamento de Marketing va a presentar la herramienta en el stand principal.

6.3.2 Contexto del Sprint y estado actual:

- **Fase del proyecto:** Estamos en el Sprint 5 de 6.

- **Cronograma:** Quedan exactamente 3 semanas para la feria. La ruta crítica actual marca que nos quedan 2 semanas de desarrollo (integración del servidor de correos electrónicos para enviar alertas) y 1 semana entera dedicada exclusivamente a QA (Pruebas de Calidad) y ciberseguridad.
- **Presupuesto:** El proyecto total era de 50.000€. Nos quedan 12.000€ de presupuesto restante, que ya están asignados para pagar las horas del equipo en estas últimas 3 semanas.
- **Recursos actuales:** 2 Desarrolladores Backend (ahora mismo configurando los servidores de email), 1 Desarrollador Frontend y 1 Ingeniero de Calidad (QA).

6.3.3 La solicitud de cambio (Change Request):

- **Descripción del cambio:** El Director de Marketing acaba de salir de una reunión con un cliente clave y exige detener la integración de las alertas por correo electrónico. Solicita que, en su lugar, integremos la **API de WhatsApp Business** para que el portal envíe notificaciones push automáticas a los móviles de los clientes cuando un palet cambie de estado.
- **Razón del cambio:** Nuestro principal competidor lanzó ayer una app con notificaciones móviles. Marketing asegura que si presentamos nuestro portal en la feria enviando solo "correos electrónicos tradicionales", la herramienta nacerá muerta y pareceremos obsoletos ante la industria. Es una decisión estratégica de negocio.

6.3.4 Datos técnicos extra para el análisis:

Integrar la API de WhatsApp no estaba en el alcance. Requiere dar de alta un número en Meta (puede tardar 3-5 días en verificarse), reestructurar los *webhooks* del servidor (estimado en 1.5 semanas extra de trabajo para los dos Backend) y obligaría al ingeniero de QA a rehacer todo el plan de pruebas de estrés. El coste de la licencia de la API y las horas extra de desarrollo sumarían unos 6.500€ no previstos.

6.4 Respuesta de la Gema

Lo primero que devuelve mi gema es el resumen ejecutivo;

La sustitución del sistema de emails por WhatsApp Business API añade una carga técnica crítica que retrasa el lanzamiento 1.5 semanas, excediendo la fecha de la feria internacional. El cambio incrementa el coste en 6.500€, agotando el margen de maniobra financiero. Se requiere una decisión inmediata entre ampliar recursos o sacrificar la fase de QA para cumplir con el hito comercial.

Además, también proporciona una matriz de impacto;

Tabla 6: Respuesta Gema Change Request

Indicador	Estado Actual (Base)	Con Cambio (Propuesto)	Diferencia / Impacto
Presupuesto (Total)	50.000 €	56.500 €	+ 6.500 € (+13%)
Presupuesto Restante	12.000 €	5.500 € (Déficit)	- 112.5% del remanente
Cronograma (Lanzamiento)	3 semanas (Sprint 6)	4.5 semanas	+ 1.5 semanas (+50%)
Ruta Crítica	Desarrollo Email + QA	Alta Meta + Webhooks + QA	Bloqueo por validación Meta

Con esto llegamos al aviso de estado actual de dicho cambio a implementar;

⚠ ALERTA DE RIESGO: IMPACTO CRÍTICO

Nivel de Riesgo: Muy Alto. El impacto presupuestario (+13%) y el retraso en el cronograma (+50% sobre el tiempo restante) superan los umbrales de seguridad. Existe un riesgo real de llegar a la feria "Logistics & Automation" con una herramienta inestable o incompleta debido a la compresión de la fase de QA.

Figura 14: Respuesta Gema Change Request

Para finalizar, la gema nos proporciona tanto Estrategias de Mitigación Prescriptivas como un Borrador de Correo Electrónico:

6.4.1 Estrategias de mitigación prescriptivas

- **Opción A: Aceleración con Recursos Externos (Fast-Tracking)** Contratar un desarrollador Senior Freelance con experiencia previa en APIs de Meta para que trabaje exclusivamente en la integración de webhooks durante 7 días. Esto permitiría que los 2 Backend actuales sigan con la lógica de negocio.
 - *Ventaja:* Mantiene la calidad del QA.
 - *Inconveniente:* Aumenta el déficit presupuestario en aproximadamente 3.000€ adicionales.

- **Opción B: Reducción de Alcance Operativo (MVP para Feria)** Integrar WhatsApp solo para un estado de palet específico (ej. "Entregado") en lugar de toda la trazabilidad, y posponer la seguridad de nivel industrial para el Sprint 7 (post-feria).
 - *Ventaja:* Se llega a la fecha de la feria con la funcionalidad visualmente operativa.
 - *Inconveniente:* El portal saldrá con vulnerabilidades de seguridad conocidas o fallos en condiciones de estrés.

6.4.2 Borrador de correo electrónico

Asunto: Urgente: Evaluación de impacto - Integración de Notificaciones WhatsApp (Proyecto Track&Trace)

Estimado [Nombre del Cliente/Director]:

Espero que se encuentre bien. Tras recibir la solicitud estratégica para sustituir las alertas de correo electrónico por la integración de **WhatsApp Business API**, nuestro equipo de ingeniería ha realizado un análisis de impacto técnico y financiero.

Dada la cercanía de la feria "Logistics & Automation", debemos informarle que esta modificación implica cambios significativos en el plan original:

1. **Impacto en Fecha:** La integración añade 1.5 semanas a la ruta crítica, lo que situaría el cierre técnico después del inicio de la feria.
2. **Impacto Financiero:** El coste se incrementa en 6.500€ debido a licencias y horas de desarrollo extraordinarias.
3. **Riesgo de QA:** Para intentar llegar a la fecha del evento, nos veríamos obligados a reducir el tiempo de pruebas de ciberseguridad.

Estamos alineados con la visión de Marketing de evitar la obsolescencia frente a la competencia, pero necesitamos su validación sobre cuál de las estrategias de mitigación propuestas (ampliación de presupuesto o reducción de alcance en pruebas) prefiere seguir para asegurar el éxito del lanzamiento.

Quedamos a la espera de su respuesta para proceder con el cambio o mantener el plan actual.

Atentamente,

Gestor de Change Requests *Project Management Office*

7 ANÁLISIS COMPARATIVO DE LAS GEMAS DESARROLLADAS

Una vez diseñadas y validadas las cinco Gemas a lo largo de los capítulos anteriores, conviene detenerse y mirarlas en conjunto. No todas resuelven el mismo tipo de problema ni aportan lo mismo al ingeniero, así que para compararlas de forma justa he definido seis criterios de evaluación, puntuados del 1 al 5. El primero, la complejidad del prompt, mide cuánta ingeniería de instrucciones hay realmente detrás de cada Gema: no es lo mismo asignarle un rol y unas reglas que dotarla de lógica condicional, fases o capacidad de interpretar datos tabulados. El valor para el ingeniero valora el ahorro real de tiempo y de trabajo administrativo que supone en el día a día. El rigor técnico comprueba si la respuesta que genera tiene verdadero criterio de ingeniería industrial o se queda en algo genérico que cualquier chat podría producir. El control humano es, en mi opinión, el criterio más importante en un entorno industrial: mide hasta qué punto la Gema respeta el principio de que la IA asiste, pero nunca decide sola. La reproducibilidad evalúa si el experimento podría repetirse y verificarse por un tercero, algo condicionado por la confidencialidad de algunos casos. Por último, la originalidad premia las soluciones que se alejan de lo evidente. La siguiente tabla recoge la valoración final de cada Gema según estos seis ejes.

Las puntuaciones se han contrastado con un modelo de Inteligencia Artificial independiente (Claude, de Anthropic) para evitar que el propio modelo sobre el que se han construido las Gemas (Gemini) evalúe su rendimiento.

Tabla 7: Comparativa de Gemas

	Gema (rol)	Complejidad del prompt (1-5)	Valor para el ingeniero (1-5)	Rigor técnico industrial (1-5)	Control humano-seguridad (1-5)	Reproducibilidad (1-5)	Originalidad (1-5)
1	Agile Coach	4	4	3	2	4	2
2	Analista Técnico de Refinamiento	4	5	5	3	4	4
3	Scrum Master Analítico	5	5	4	5	3	5
4	Analista de KPIs	4	4	4	3	5	3
5	Experto en Change Management	5	4	4	4	3	4

De la lectura conjunta de la tabla se extraen varias conclusiones que van más allá de las cifras concretas de cada celda.

La Gema más completa es, sin discusión, la tercera: el Radar de Riesgos. Es la que mejor combina los tres ejes que de verdad importan en este trabajo, liderando en complejidad del prompt, en control humano y en originalidad. Su diseño en fases detecta el riesgo, lo clasifica con un semáforo, se frena y pregunta al ingeniero,

y solo ejecuta tras la decisión humana. Esto es lo que la convierte en una herramienta industrialmente fiable y no en un simple generador de texto. Si tuviera que señalar una sola Gema como buque insignia del proyecto, sería esta.

La segunda Gema, la de Gestión y Control, es la que mejor demuestra rigor técnico real. Cuando traduce un deseo comercial ambiguo en parámetros físicos concretos exigiendo un factor de servicio mayor o igual a 1,5, calculando el par motor con carga máxima o limitando la flecha de los perfiles a menos de 2 mm deja claro que el prompt maestro que hay detrás está bien construido. Esa salida no la produce un chat genérico, y es la mejor prueba de que la ingeniería de prompts funciona cuando se hace con criterio.

Desde el punto de vista del método, la Gema más sólida es la cuarta, el Analista de KPIs, aunque sea la menos vistosa. Su validación se acerca a un diseño experimental: se le entregan los datos crudos de un Sprint y se comprueba si llega al mismo diagnóstico que las métricas ya calculadas. Por eso es la que mejor puntúa en reproducibilidad, ya que su resultado puede contrastarse de forma objetiva.

Pero lo más relevante no es ninguna Gema por separado, sino el patrón que recorre las cinco y que da unidad a todo el trabajo: la idea de que la IA asiste, pero no decide. Casi todas las herramientas, en lugar de imponer una solución, se detienen en algún punto para devolver el control al ingeniero. Este enfoque, conocido como *human-in-the-loop*, no es un detalle menor: es justo lo que un entorno industrial exige de la Inteligencia Artificial, donde un error de criterio no se corrige borrando una línea de código, sino que puede costar un motor mal pedido, un retraso en la cadena de suministro o un riesgo para un operario. Que las cinco Gemas compartan esta filosofía demuestra que el valor de combinar Agile e IA no está en sustituir al ingeniero, sino en liberarlo del trabajo repetitivo para que se centre en lo que de verdad aporta valor: la toma de decisiones técnicas.

7.1 Interpretación gráfica de la comparación

Aunque la Tabla 7 recoge toda la información de forma ordenada, leerla de manera horizontal mientras comparas Gema a Gema en cada criterio puede hacerse un poco lento. Por eso he decidido representar los mismos datos en forma de gráfico de radar, una herramienta muy utilizada en análisis multicriterio precisamente por su capacidad de mostrar, de un solo vistazo, el perfil completo de cada elemento evaluado sobre todos sus ejes a la vez, dejando ver al instante dónde destaca y dónde flojea cada uno. Para que la comparación sea lo más clara posible, he asignado a cada Gema un color y una forma de marcador distintos, de modo que se diferencien con facilidad incluso si el documento se imprime en blanco y negro. El trazo de la Gema 3 se ha mantenido ligeramente más grueso porque, como ya se ha argumentado, es la herramienta más completa del conjunto y conviene que su perfil destaque visualmente sobre el resto.

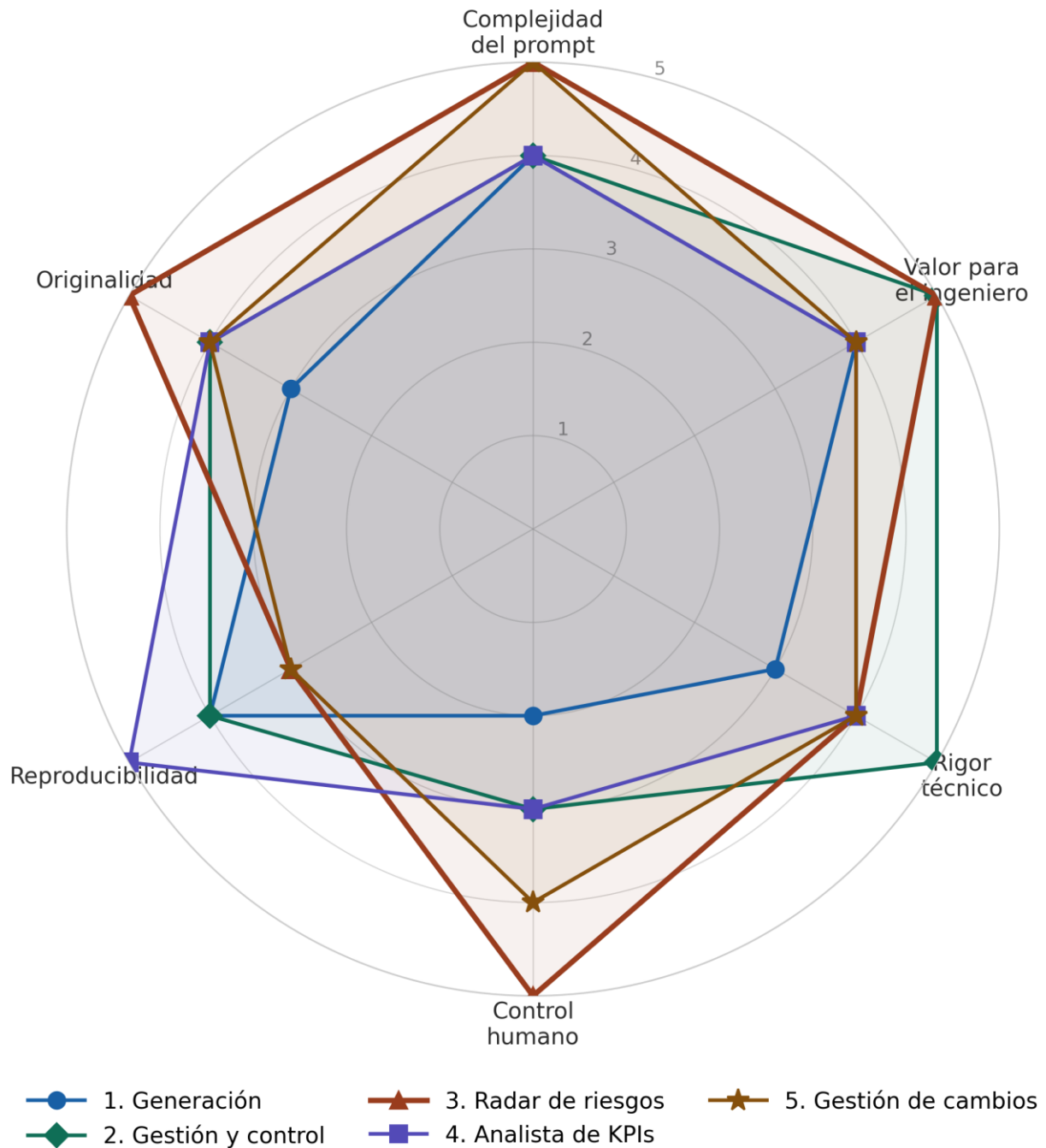


Figura 15: Perfil comparativo de las cinco Gemas según los seis criterios de evaluación

8 CONCLUSIÓN

La realización de este Trabajo de Fin de Grado permite confirmar el cumplimiento del objetivo general planteado al inicio: demostrar que la metodología Agile, lejos de ser exclusiva del desarrollo de software, es un marco de trabajo válido para la organización industrial, y que su combinación con la Inteligencia Artificial generativa abre una vía real para modernizar la gestión de proyectos en este sector. A lo largo del trabajo se ha argumentado por qué el modelo predictivo tradicional en cascada se queda corto frente a la volatilidad actual de los mercados y las cadenas de suministro, y se ha mostrado cómo la división del proyecto en Sprints permite al equipo chocar pronto con los imprevistos físicos, minimizando el impacto económico de los errores.

El verdadero valor diferencial del trabajo, sin embargo, está en haber atacado el "lado oscuro" de la agilidad: la carga administrativa y documental que asfixia a los roles de Product Owner y Scrum Master. Mediante ingeniería de prompts [21], se ha diseñado y validado un conjunto de cinco asistentes virtuales (Gemas) sobre el modelo Gemini que cubren el ciclo de vida completo del proyecto. La principal conclusión metodológica del estudio es que la Inteligencia Artificial, en este contexto, no debe entenderse como un generador automático de texto, sino como un filtro antierrores acoplado a los estándares de la ingeniería, en concreto a los criterios INVEST, la escala de Fibonacci y las prácticas habituales de gestión de proyectos. Bajo esas restricciones, un modelo de lenguaje generalista pasa a comportarse con el rigor conceptual de un gestor experimentado.

El análisis individual y conjunto de las cinco herramientas, recogido en la tabla comparativa del capítulo anterior, deja ver una estructura coherente con las fases del proyecto. La primera Gema, de Generación de Historias de Usuario, ha demostrado ser un puente útil entre la ambigüedad de los requisitos comerciales y la estructuración formal del backlog. La segunda, de Gestión y Control, ha actuado como escudo técnico del equipo: el límite estricto de ocho puntos en la escala de Fibonacci ha forzado la división de "historias monstruo" en subtarefas modulares y testeables, traduciendo los deseos del cliente en parámetros físicos concretos (cadencia, dimensiones, motorización). La tercera, el Radar de Riesgos, es la herramienta más completa del conjunto: clasifica los bloqueos del Daily mediante un código de semáforo y, sobre todo, incorpora un freno interactivo que la obliga a detenerse y consultar al ingeniero antes de actuar. La cuarta, el Analista de KPIs, automatiza el cálculo de métricas como la velocidad del equipo o el Say/Do Ratio, pero su aportación más interesante es la generación de un KPI contextual adaptado a las incidencias reales de cada Sprint. Y la quinta, el Gestor de Change Requests, combate el Scope Creep evaluando el impacto de una solicitud sobre el cronograma y proponiendo alternativas de mitigación junto con la comunicación necesaria para alinear a las partes implicadas.

Detrás de estas cinco herramientas hay una tesis que recorre todo el trabajo y que conviene dejar explícita: el avance de la IA en la organización industrial no busca sustituir al ingeniero, sino potenciar su criterio. Las instrucciones maestras de las Gemas se han diseñado deliberadamente para incluir pausas, preguntas obligatorias y puntos de control donde la decisión final vuelve al usuario. La responsabilidad ética, contractual y de seguridad de un proyecto físico no puede delegarse en un modelo probabilístico, y los casos de validación presentados incluidos los escenarios de mayor fricción técnica confirman que la Gema funciona como asesor que ordena el caos informativo y propone alternativas, pero la firma y la elección del camino siguen siendo del ingeniero humano.

Reconocer las limitaciones del estudio es parte del trabajo. La principal es que la validación se ha realizado sobre casos simulados o anonimizados, condicionados por la confidencialidad de mi experiencia durante las prácticas en Accenture; aunque la estructura de los datos replica la de un entorno real, no equivale a una implantación en planta. A partir de aquí se abren tres líneas futuras claras. La primera, la integración nativa de las Gemas con las herramientas que ya se usan en la industria, sustituyendo el actual flujo manual de copiar y pegar por conexiones directas mediante APIs con plataformas de gestión de proyectos y sistemas corporativos. La segunda, la incorporación de técnicas avanzadas de ingeniería de prompts, como Few-Shot Prompting [21] o Chain-of-Thought [22], para afinar la precisión de la IA al estimar tareas en disciplinas mecánicas o eléctricas, más alejadas del software puro. Y la tercera, la validación empírica en una planta real durante al menos seis meses,

midiendo a lo largo de varios Sprints consecutivos variables como el tiempo de ciclo, la reducción del lead time o la evolución del Say/Do Ratio. Solo una prueba de ese tipo permitirá cuantificar de forma rigurosa el retorno real de la agilidad asistida por IA en el tejido industrial, que es la pregunta natural que abre este trabajo y que espero poder responder, en algún momento, con datos de campo.

REFERENCIAS

- [1] K. Beck et al., "Manifiesto for Agile Software Development," 2001. [En línea]. Disponible: <https://agilemanifesto.org/>
- [2] H. Takeuchi e I. Nonaka, "The New New Product Development Game," Harvard Business Review, vol. 64, n.º 1, pp. 137-146, ene.-feb. 1986.
- [3] K. Schwaber y J. Sutherland, The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game, nov. 2020. [En línea]. Disponible: <https://scrumguides.org/>
- [4] Project Management Institute, A Guide to the Project Management Body of Knowledge (PMBOK Guide), 7.ª ed. Newtown Square, PA: PMI, 2021. ISBN: 978-1-62825-664-2.
- [5] M. Cohn, User Stories Applied: For Agile Software Development. Boston, MA: Addison-Wesley, 2004. ISBN: 978-0-321-20568-1.
- [6] R. Jeffries, "Essential XP: Card, Conversation, Confirmation," XProgramming.com, ago. 2001. [En línea]. Disponible: <https://ronjeffries.com/xprog/articles/expcardconversationconfirmation/>
- [7] B. Wake, "INVEST in Good Stories, and SMART Tasks," XP123, ago. 2003. [En línea]. Disponible: <https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>
- [8] T. Ohno, Toyota Production System: Beyond Large-Scale Production. Cambridge, MA: Productivity Press, 1988. ISBN: 978-0-915299-14-0.
- [9] D. J. Anderson, Kanban: Successful Evolutionary Change for Your Technology Business. Sequim, WA: Blue Hole Press, 2010. ISBN: 978-0-9845214-0-1.
- [10] J. P. Womack y D. T. Jones, Lean Thinking: Banish Waste and Create Wealth in Your Corporation. Nueva York, NY: Simon & Schuster (Free Press), 1996. ISBN: 978-0-684-81035-5.
- [11] J. P. Womack, D. T. Jones y D. Roos, The Machine That Changed the World: The Story of Lean Production. Nueva York, NY: Rawson Associates / Macmillan, 1990.
- [12] J. K. Liker, The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer. Nueva York, NY: McGraw-Hill, 2004. ISBN: 978-0-07-139231-0.
- [13] K. Beck, Extreme Programming Explained: Embrace Change. Reading, MA: Addison-Wesley, 2000. ISBN: 978-0-201-61641-5.
- [14] M. Poppendieck y T. Poppendieck, Lean Software Development: An Agile Toolkit. Boston, MA: Addison-Wesley Professional, 2003. ISBN: 978-0-321-15078-3.
- [15] J. Sutherland, Scrum: The Art of Doing Twice the Work in Half the Time. Nueva York, NY: Crown Business, 2014. ISBN: 978-0-385-34645-0.
- [16] J. Highsmith, Agile Project Management: Creating Innovative Products. Boston, MA: Addison-Wesley Professional, 2004. ISBN: 978-0-321-21977-0.
- [17] A. Bahi, J. Gharib y Y. Gahi, "Integrating Generative AI for Advancing Agile Software Development and Mitigating Project Management Challenges," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 15, n.º 3, pp. 50-58, 2024. doi: 10.14569/IJACSA.2024.0150306.
- [18] S. Russell y P. Norvig, Artificial Intelligence: A Modern Approach, 4.ª ed. Pearson, 2021.
- [19] A. Vaswani et al., «Attention Is All You Need», Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017.
- [20] J. Wei et al., «Emergent Abilities of Large Language Models», Transactions on Machine Learning Research, ago. 2022.
- [21] P. Liu et al., «Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing», ACM Computing Surveys, vol. 55, n.º 9, pp. 1-35, 2023.

- [22] J. Wei et al., «Chain-of-Thought Prompting Elicits Reasoning in Large Language Models», NeurIPS, 2022.
- [23] L. Weng, «LLM Powered Autonomous Agents», Lil'Log, 23 jun. 2023. [En línea]. Disponible en: <https://lilianweng.github.io/posts/2023-06-23-agent/>
- [24] Google, Gemini (modelo de lenguaje de gran escala). Asistente de IA generativa utilizado para la creación de las Gemas descritas en los capítulos 3, 4 y 5. Google LLC, 2024-2025. [En línea]. Disponible: <https://gemini.google.com/>
- [25] Anthropic, Claude (modelo de lenguaje de gran escala). Asistente de IA generativa utilizado como apoyo durante la elaboración del presente trabajo. Anthropic PBC, 2024-2025. [En línea]. Disponible: <https://claude.ai/>
- [26] Google, NotebookLM. Asistente de investigación basado en IA utilizado como apoyo durante la elaboración del presente trabajo. Google LLC, 2024-2025. [En línea]. Disponible: <https://notebooklm.google.com/>