

Fundamentos de la Ciencia de Datos: PEC3

Víctor Suesta Arribas

Diciembre 2025



Pregunta 1

- Define formalmente el proyecto según las metodologías PMBOK, PRINCE2 e ISO21500.
- Haz una comparativa en detalle entre el Ciclo de Vida del Producto y el Ciclo del Proyecto. Ejemplifica en el proyecto cada una de las fases.

1.1. Definición formal del proyecto según PMBOK®, PRINCE2 e ISO 21500

El proyecto que utilizo como caso es DomoticAI, una solución basada en un agente inteligente para integrar sensores/actuadores y fuentes externas de datos con el objetivo de optimizar eficiencia energética y confort en entornos de domótica e inmótica.

Desde PMBOK®, un proyecto se entiende como un esfuerzo temporal orientado a crear un producto, servicio o resultado único. En DomoticAI lo interpreto así porque el trabajo tiene un inicio y un cierre (se persigue entregar un MVP verificable) y porque el resultado es singular por la combinación de interoperabilidad, integración de datos y toma de decisiones autónoma. Además, el material insiste en que una idea se convierte en proyecto cuando se concretan objetivos, un marco temporal y una estimación inicial de recursos, costes y tiempo; en este caso, esto se refleja al definir el alcance del MVP, planificar iteraciones y asignar recursos de desarrollo e infraestructura.

En PRINCE2, el proyecto se describe como una organización temporal creada para entregar uno o más productos de negocio apoyados en un plan o caso de negocio. Esta perspectiva encaja bien con DomoticAI porque obliga a justificar la entrega en términos de valor: el producto se plantea para generar beneficios medibles (ahorro, confort, sostenibilidad) y su lógica de negocio queda reflejada en los elementos trabajados en el BMC (segmentos, propuesta de valor y fuentes de ingresos).

Por último, ISO 21500 define el proyecto como un conjunto único de procesos coordinados y controlados, con inicio y fin, para alcanzar un objetivo cumpliendo requisitos y restricciones (tiempo, coste y recursos). En DomoticAI esto me ayuda a hacer explícito que, además de desarrollar la solución, hay que gestionarla bajo limitaciones reales (calendario, presupuesto e incluso disponibilidad de integraciones o proveedores), manteniendo control sobre el trabajo para llegar a un entregable aceptable.

En conjunto, las tres definiciones coinciden en lo esencial (temporalidad y resultado único), pero aportan matices complementarios: PMBOK® enfatiza la naturaleza de “esfuerzo temporal”, PRINCE2 refuerza la orientación a producto de negocio e ISO 21500 obliga a explicitar restricciones y control del proyecto.

1.2. Comparativa entre Ciclo de Vida del Producto y Ciclo de Vida del Proyecto (con ejemplo aplicado)

El ciclo de vida del producto se entiende como el conjunto de fases necesarias para producir un producto y mantenerlo, hasta llegar a su retirada. El material lo describe como un recorrido secuencial que suele arrancar en la concepción, continúa en el desarrollo y se prolonga en el uso/presencia en el mercado, donde se distinguen etapas típicas como introducción, crecimiento, madurez y declive. Aplicado a DomoticAI, la concepción se refleja en la definición de la propuesta y su encaje como solución de valor; el desarrollo se corresponde con la construcción del prototipo y la lógica del agente; y el uso se materializa en la implantación progresiva (pilotos), la escalabilidad por integraciones y la evolución del producto cuando ya está operativo.

En cambio, el ciclo de vida del proyecto hace referencia a las fases por las que pasa el trabajo desde su inicio hasta su cierre para producir un entregable. El material remarca que, a lo largo de ese ciclo, los costes y recursos suelen ser bajos al inicio, máximos en fases intermedias y decrecen al final; que la incertidumbre y el riesgo son mayores al principio; y que el coste de los cambios aumenta con el avance del proyecto, por lo que es especialmente relevante implicar interesados en etapas tempranas. En DomoticAI, esto se observa con bastante claridad: al inicio se define el alcance del MVP y se reduce incertidumbre; en fases intermedias se concentra el desarrollo e integración (donde el esfuerzo es mayor); y al final se valida el entregable y se cierra el trabajo de proyecto.

La diferencia práctica que extraigo es que el producto puede vivir durante años y evolucionar en el mercado, mientras que el proyecto es necesariamente temporal y termina al entregar un resultado. Por eso, el material también señala que los proyectos pueden “aparecer” a lo largo de la vida del producto: se puede ejecutar un proyecto para lanzarlo, otro para ampliarlo o incluso uno para rediseñarlo durante la madurez. En DomoticAI, el proyecto inicial sería el del MVP, pero en el ciclo del producto aparecerían después proyectos adicionales para nuevas integraciones, mejoras o despliegues en otros segmentos.

Referencias utilizadas: [1].

Fase	Ciclo de vida del producto (DomoticAI)	Ciclo de vida del proyecto (MVP DomoticAI)
Concepción / Idea	Defino la propuesta de valor (ahorro, confort, sostenibilidad) y el encaje domótica/inmótica.	Inicio el proyecto: objetivo del MVP, alcance inicial y criterios básicos de éxito.
Planificación	Aterrizo qué versiones del producto tienen sentido (básica vs pro, integraciones prioritarias).	Planifico entregables y trabajo: backlog del MVP, estimación de esfuerzo y recursos, riesgos principales.
Desarrollo / Construcción	Desarrollo la base del producto (flujo, pantallas, lógica del agente en prototipo) y preparo la arquitectura de integraciones.	Ejecuto el plan: desarrollo e integración de fuentes/sensores en el MVP (máxima carga de trabajo).
Validación / Control	Verifico que el producto cumple lo esperado para el usuario (confianza, explicabilidad de decisiones, ajustes de autonomía).	Monitorizo y controlo: reviso avances, gestiono cambios, reduzco incertidumbre con feedback temprano.
Lanzamiento / Uso	Despliego en pilotos y lo pongo en operación; empiezo a medir ahorro y mejoras.	Cierro el proyecto: acepto el MVP, documento resultados y cierro el trabajo del proyecto.
Evolución	El producto sigue vivo: nuevas integraciones, mejoras y posibles versiones para otros segmentos.	Surgen nuevos proyectos: ampliaciones, rediseños o despliegues para hoteles/empresas.

Cuadro 1: Comparación explícita (fase a fase) entre el ciclo de vida del producto y el ciclo de vida del proyecto aplicada a DomoticAI.

Pregunta 2

- Define el proyecto dentro del Marco de Clasificación Cynefin.
- Siguiendo la tipología de R. K. Wysocki, describe las características de este proyecto.
- Para decidir quién se responsabilizará del proyecto: ¿qué criterios debemos tener en cuenta?, ¿qué responsabilidades y competencias se le asignarán?, ¿qué habilidades debe tener?
- ¿Quiénes serían los stakeholders o interesados en el proyecto?

2.1. Definición del proyecto dentro del Marco de Clasificación Cynefin

Yo entiendo el marco de Cynefin (tal como se presenta en los materiales) como una forma de clasificar proyectos según el grado de incertidumbre de los requerimientos y de la tecnología, y también según cómo se conoce (o no) la relación causa-efecto. En los contextos simples la relación causa-efecto es clara y repetible; en los complicados existe pero no es evidente y suele requerir experiencia; en los complejos la relación causa-efecto no es obvia y los resultados no se pueden predecir con facilidad; y en los caóticos no se aprecia una relación causa-efecto y el foco pasa a la gestión de la crisis.

En el caso de DomoticAI, yo situaría el proyecto principalmente en el dominio complejo. Aunque el objetivo general es entendible (gestionar de forma autónoma energía y confort), el comportamiento del sistema depende de elementos interrelacionados: sensores y actuadores de distinta naturaleza, preferencias humanas, condiciones externas (por ejemplo, meteorología u otras fuentes abiertas) y, además, decisiones en tiempo real. En este tipo de proyectos, yo destaco lo que indica el material: que los requerimientos y la tecnología suelen estar “ligeramente definidos”, que el conocimiento se va generando durante el proyecto y que se avanza mediante iteraciones con un resultado en cada iteración y evaluación rápida para guiarse en un entorno de alta incertidumbre. Esto encaja bastante bien con una solución software que integra múltiples componentes y necesita validarse progresivamente con usos reales (domótica e inmótica). Además, el propio material pone el desarrollo de software como ejemplo típico de proyecto complejo, lo cual me refuerza esta clasificación.

2.2. Tipología de R. K. Wysocki y características del proyecto

Yo tomo el planteamiento de Wysocki como una clasificación de proyectos según dos variables: la claridad del objetivo y la claridad de la solución. A partir de ahí se obtiene una matriz de cuatro cuadrantes: proyectos tradicionales (Q1), ágiles (Q2), extremos (Q3) y emertxe (Q4).

Aplicándolo a DomoticAI, yo lo ubicaría en Q2 (proyecto ágil): el objetivo es relativamente claro (conseguir un agente que integre datos y actúe para optimizar eficiencia y confort), pero la solución no está completamente definida desde el inicio porque hay decisiones técnicas y de producto que dependen de aprendizaje incremental: qué integraciones priorizar, cómo ajustar el nivel de autonomía o cómo validar que las acciones del agente son aceptables en contextos distintos. Yo me apoyo en lo que describe el material: que, en proyectos ágiles, el cliente puede tener clara la necesidad y lo que quiere conseguir, pero se desconoce cómo lograrlo, y por eso es especialmente relevante la implicación del cliente en el desarrollo de la solución.

Por contraste, yo no lo situaría en Q1 porque aquí la solución no es evidente desde el inicio; tampoco en Q3 porque el objetivo no está “totalmente” indefinido (sí sabemos qué queremos conseguir a nivel de valor); y tampoco en Q4 porque

no partimos de una solución cerrada a la que luego se le busca aplicación.

2.3. Responsable del proyecto: criterios, responsabilidades, competencias y habilidades

Según los materiales, yo entiendo al director de proyecto como la persona responsable de gestionar el proyecto, cumplir los objetivos y guiarlo hacia el éxito desde la planificación hasta el cierre. Además, yo recojo la evolución del rol que se menciona (cambio de modelo): no tiene por qué ser “quien más sabe” técnicamente, sino un perfil más facilitador, con visión integral, que coordina, elimina impedimentos y gestiona aspectos como riesgos, calidad e interesados.

Con esa base, los criterios que yo considero más razonables para seleccionar a la persona responsable en DomoticAI son, primero, que tenga una visión global para coordinar un proyecto con varios actores (usuarios, instaladores/partners, proveedores de tecnología y datos), y, segundo, que pueda alinear el trabajo con el valor de negocio (ahorro, confort y sostenibilidad). Yo entiendo este equilibrio a través del Triángulo de Talento del PMI, que combina dirección técnica de proyectos, gestión estratégica y de negocios, y liderazgo. En un proyecto como este, para mí esa combinación es especialmente útil porque se necesita método para gestionar restricciones y entregables, orientación a beneficios (propuesta de valor y adopción) y capacidad de liderazgo para alinear a equipos e interesados.

Respecto a responsabilidades, yo tomo lo que indica el material: identificar requerimientos y riesgos, desarrollar el plan de proyecto y subplanes, guiar el proyecto para cumplir objetivos e identificar y gestionar necesidades y expectativas de los interesados. En DomoticAI yo lo traduciría, por ejemplo, en delimitar bien el alcance inicial (MVP), ordenar prioridades de integración y asegurar comunicación continua con los interesados, especialmente al principio, porque el propio material indica que la incertidumbre es mayor al inicio y que el coste de los cambios aumenta a medida que el proyecto avanza; por eso yo considero que conviene implicar interesados de forma intensa desde las etapas tempranas.

Por último, en habilidades y competencias, yo uso como referencia el estándar PMCD citado en el texto, que agrupa competencias personales como comunicación, liderazgo, dirección, habilidades cognitivas, eficacia y profesionalidad. En este caso yo destacaría comunicación (adaptar el mensaje a audiencias distintas), liderazgo (mantener relaciones eficaces y responsabilizarse de entregar), habilidades cognitivas (visión holística y resolución de problemas) y capacidad de influencia/negociación, porque la coordinación con múltiples partes externas es una condición práctica del proyecto.

2.4. Stakeholders o interesados del proyecto

Según los materiales, yo entiendo por interesado (stakeholder) a cualquier individuo, grupo u organización que puede afectar, verse afectado o percibirse como afectado por una decisión, actividad o resultado del proyecto. Además, yo incorporo la idea de que su identificación y gestión es un proceso continuo durante el ciclo de vida del proyecto.

En DomoticAI, yo incluiría como interesados internos al patrocinador (quien autoriza y/o financia), al director del proyecto y al equipo. En el lado externo, yo identifico claramente a los clientes y usuarios finales (hogares; hoteles/coworkings/oficinas en inmótica), así como a actores que condicionan el despliegue y la interoperabilidad: instaladores, constructoras/ingenierías que integran la solución en proyectos, fabricantes o proveedores de hardware y servicios de integración, y proveedores de datos externos cuando se utilicen fuentes adicionales. También me parece razonable considerar interesados como proveedores, competidores y, en función del despliegue, entidades reguladoras o asociaciones relevantes, tal y como ejemplifica el material.

Desde una perspectiva de gestión, para mí lo importante es que esta lista no sea “estática”: al tratarse de un proyecto con incertidumbre, el peso relativo de cada interesado puede cambiar, y yo considero que el director del proyecto debe ir ajustando la estrategia para maximizar impactos positivos y evitar bloqueos o resistencias, coherente con la idea de gestión continua de interesados.

Referencias utilizadas: [1, 2].

Pregunta 3

Para caracterizar DomoticAI con CRISP-DM, tomo el proceso como un ciclo de seis fases (Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation y Deployment), donde la secuencia no es estricta y es habitual volver atrás entre fases cuando aparecen hallazgos o limitaciones nuevas.

3.1. Business Understanding

En esta fase, mi prioridad es traducir la idea “optimizar confort y eficiencia en domótica/inmótica” a objetivos de negocio verificables y a criterios de éxito claros, evitando que el proyecto se convierta en una exploración técnica sin dirección. CRISP-DM plantea precisamente que, antes de trabajar con datos, conviene involucrar a las personas clave, documentar expectativas y acordar criterios de éxito desde la perspectiva de negocio. En DomoticAI lo concreto en beneficios y restricciones: reducción de consumo energético sin degradar confort, disminución de incidencias operativas y mejora de la experiencia del usuario; y, a la vez, limitaciones de presupuesto, plazos, hardware disponible e integraciones realistas.

Además, en Business Understanding establezco el puente hacia lo analítico: defino qué problema de data mining/modelado es relevante (por ejemplo, predicción, detección de anomalías u optimización), qué horizonte temporal necesito (minutos, horas, días) y qué significa “éxito” tanto en términos de negocio como en términos técnicos, porque CRISP-DM insiste en que la metodología requiere convertir objetivos de negocio en objetivos de minería/modelado y criterios de evaluación.

3.2. Data Understanding

En Data Understanding paso de la intención a la realidad: identifico qué datos existen, cómo se capturan y cuál es su calidad, porque esta fase se define como el momento de “mirar de cerca” los datos disponibles y explorar su contenido para evitar problemas inesperados en la preparación, que suele ser la fase más costosa. En DomoticAI esto implica inventariar fuentes internas (sensores de temperatura/humedad/CO₂, consumos, estados de actuadores, horarios/ocupación si existe, logs del sistema) y fuentes externas potenciales (meteorología, tarifas eléctricas), además de verificar accesos y restricciones.

A nivel práctico, analizo series temporales y eventos: frecuencia real de muestreo, huecos, lecturas imposibles, outliers, latencias y coherencia entre fuentes (por ejemplo, si un cambio de consigna se refleja en consumo). CRISP-DM recomienda, en esta fase, describir tipos de valores y esquemas de codificación, calcular estadísticas básicas y usar exploración para generar hipótesis y preparar transformaciones posteriores; en DomoticAI esa exploración guía decisiones como el re-muestreo, la definición de ventanas y la selección de atributos.

3.3. Data Preparation

En Data Preparation convierto los datos “tal como llegan” en un dataset utilizable por modelos y por reglas operativas. CRISP-DM enfatiza que la preparación suele absorber una parte muy relevante del esfuerzo total y que incluye tareas como seleccionar, integrar, derivar atributos, tratar faltantes y preparar particiones de entrenamiento/prueba. En DomoticAI lo interpreto como un pipeline reproducible que unifica fuentes heterogéneas en un eje temporal común, con una semántica coherente.

De forma concreta, preparo (i) sincronización temporal (alinear sensores y eventos), (ii) limpieza (valores físicamente imposibles, sensores defectuosos, imputación o marcaje explícito de faltantes), (iii) integración (unir por identificadores de dispositivo/estancia y por tiempo), y (iv) construcción de variables: retardos, medias móviles, gradientes, variables de calendario, y agregaciones por zona o edificio. Cuando incorporo datos externos, verifico también consistencia de formatos y posibles problemas de unión, porque CRISP-DM remarca que la integración y el formateo para el modelado requieren comprobar que los merges se han realizado correctamente y que el formato final se ajusta a las técnicas elegidas.

3.4. Modeling

En Modeling construyo y comparo alternativas para responder a los objetivos definidos: aquí es donde la preparación “empieza a pagar”, pero asumo desde el principio que será iterativo. CRISP-DM describe que lo habitual es ejecutar varios modelos, probar parámetros por defecto y ajustar, e incluso volver a preparación si el modelo exige transformaciones específicas; rara vez un único modelo y una única ejecución resuelven el problema de forma satisfactoria. Para DomoticAI, eso se traduce en experimentar con familias de soluciones que cubran necesidades distintas: predicción de consumo/demanda, detección de anomalías (fallos o consumos atípicos), estimación de confort o recomendación/optimización de consignas bajo restricciones.

También defino un diseño de prueba coherente con el contexto (train/test temporal, validación por periodos o por edificios) y registro decisiones: supuestos, parámetros y transformaciones hechas para cumplir requisitos del modelo. CRISP-DM insiste en documentar criterios de “bondad” y los datos sobre los que se prueban, así como en mantener trazabilidad de parámetros y resultados para poder comparar de forma rigurosa y repetir el proceso.

3.5. Evaluation

En Evaluation verifico que los resultados no solo son técnicamente aceptables, sino útiles para el negocio. CRISP-DM plantea que, tras confirmar en Modeling que los modelos cumplen criterios técnicos, debo evaluar con los criterios de éxito de negocio establecidos al inicio, incorporando a los decisores clave en la valoración. En DomoticAI esto lo traduzco a una doble lectura: métricas

técnicas (errores de predicción, estabilidad, robustez ante faltantes) y métricas de impacto (ahorro energético, mantenimiento del confort, reducción de incidencias), idealmente con pilotos controlados o comparativas antes/después.

Además, en esta fase selecciono qué modelos quedan “aprobados” para pasar a operación y qué hallazgos (findings) son accionables aunque no se traduzcan en un modelo desplegable. CRISP-DM distingue precisamente entre modelos finales y conclusiones derivadas del proceso, y pide dejarlo documentado de forma presentable y orientada a decisión.

3.6. Deployment

En Deployment convierto el resultado en valor operativo. CRISP-DM define despliegue como el uso de los nuevos insights para mejorar la organización, ya sea mediante integración formal en sistemas (por ejemplo, un modelo que genera scores consumidos por otros componentes) o mediante difusión de hallazgos que cambian decisiones. Para DomoticAI, el despliegue más natural es integrar el modelo o la lógica de decisión en el agente (edge o cloud), con una interfaz segura hacia sensores/actuadores, y con un plan de monitorización para comprobar que el rendimiento se mantiene con el tiempo.

También considero el cierre ordenado: resumen de resultados, plan de despliegue, contingencias y un informe final para patrocinio y perfiles técnicos/negocio, porque CRISP-DM subraya tanto la planificación/monitorización del despliegue como las tareas de cierre (informe final y revisión del proyecto), incluso si los resultados no fueran plenamente satisfactorios.

3.7. Gestión de interesados del proyecto (stakeholders) en el contexto CRISP-DM

En DomoticAI, gestiono a los interesados como una condición de éxito transversal al ciclo CRISP-DM: en Business Understanding, CRISP-DM recomienda involucrar al máximo de personas clave para alinear expectativas y documentar acuerdos, y en Evaluation recalca incluir a los decisores en la valoración con criterios de negocio. Esto es especialmente relevante aquí porque el proyecto afecta a perfiles con prioridades distintas: usuarios (confort y control), operación/mantenimiento (estabilidad y diagnósticos), negocio (ahorro y escalabilidad) y equipo técnico (viabilidad de integraciones).

En la práctica, lo concreto en tres hábitos: (i) al inicio, cierro definiciones operativas de “confort” y de “éxito” con patrocinio y usuarios clave; (ii) durante Data Understanding y Preparation, valido con operación y técnica los supuestos sobre calidad y significado de señales; y (iii) antes de desplegar, preparo entregables adaptados a audiencias (informe ejecutivo vs. detalle técnico) y un plan de monitorización/mantenimiento, tal como CRISP-DM sugiere al hablar de informes finales y de monitorización del despliegue.

Referencias utilizadas: [3, 4].

Pregunta 4

Profundicemos ahora en cómo sería la gestión del proyecto con metodologías ágiles:

- Razona si es beneficioso utilizar este tipo de metodologías
- Detalla una iteración o sprint, ¿qué entradas y salidas tendría?
- Define roles, artefactos y eventos involucrados

4.1. Beneficios de gestionar DomoticAI con metodologías ágiles

En DomoticAI considero beneficioso utilizar metodologías ágiles porque el proyecto se desarrolla en un entorno donde el cambio es estructural: tanto los requisitos (qué integraciones son prioritarias, qué nivel de autonomía acepta el usuario, qué métricas de confort son relevantes) como la tecnología (dispositivos, APIs, calidad de señales) pueden evolucionar durante el desarrollo. En este tipo de contexto, el marco ágil resulta coherente porque asume explícitamente que los requisitos cambian y que el valor se maximiza entregando incrementos frecuentes y validables, en lugar de intentar cerrar desde el inicio una especificación exhaustiva que después sea costosa de mantener.

Además, el material presenta una diferencia clave entre enfoques tradicionales y ágiles: los métodos tradicionales se apoyan en un control predictivo (planificación detallada a largo plazo y control estricto del cambio), mientras que los métodos ágiles se basan en control empírico (inspección y adaptación regular), con ciclos cortos y retrospectiva para mejorar y ajustar. Esta lógica empírica encaja con DomoticAI porque el aprendizaje real aparece cuando el sistema se prueba con datos y uso; por tanto, prefiero organizar el trabajo para inspeccionar resultados con frecuencia y adaptar decisiones con rapidez, manteniendo el foco en el valor entregado.

Por último, desde los valores y principios del Manifiesto Ágil, veo tres ventajas prácticas para este caso: priorizar la colaboración continua con el cliente/usuario (necesaria para fijar criterios de confort y aceptabilidad), entregar producto funcional como medida real de progreso (especialmente útil en integraciones e IoT) y responder al cambio sin perder una planificación marco. En un producto como DomoticAI, donde parte de los “requisitos” son hipótesis sobre qué aporta valor, el enfoque ágil me permite validar pronto, aprender y re-priorizar en función de evidencias.

4.2. Una iteración o sprint: entradas y salidas

Para dejarlo completamente explícito, en un sprint de DomoticAI yo distinguiría:

- **Entradas:** product backlog priorizado; objetivo preliminar de producto (qué valor busco); capacidad del equipo; definición de “terminado” (calidad mínima); datos/feedback disponibles (incluyendo incidencias del sprint anterior) y restricciones técnicas (integraciones, latencias, seguridad de actuación).
- **Salidas:** incremento funcional (potencialmente utilizable); sprint backlog actualizado con el trabajo realizado; feedback formal de revisión (aceptación/rechazo); acciones de mejora de la retrospectiva; y product backlog refinado/re-priorizado.

Para concretar la gestión, describo un sprint típico orientado a un objetivo de DomoticAI: “mejorar la predicción de consumo y su integración en el agente para ajustar consignas con seguridad”.

Al inicio del sprint, las entradas principales son el product backlog priorizado y las aclaraciones del propietario del producto sobre valor y prioridad. El material define el product backlog como la lista ordenada de todo lo que puede necesitarse en el producto y la fuente única de requisitos para cualquier cambio; además, el sprint comienza con una planificación donde el equipo define el objetivo y se compromete con un incremento al final. En mi caso, el backlog incluiría historias como: integrar un nuevo sensor o protocolo, mejorar limpieza de datos, añadir una métrica de confort, o entrenar un modelo de predicción con validación temporal.

Durante la planificación del sprint, convierto esas entradas en dos resultados de arranque: (i) un objetivo de sprint que exprese el valor buscado y (ii) un sprint backlog, entendido como el subconjunto de ítems del product backlog seleccionado para el sprint, junto con el trabajo necesario para entregar el incremento. En DomoticAI, ese sprint backlog suele mezclar tareas de datos (alineación temporal, control de calidad), tareas de modelado (entrenamiento/selección de modelo) y tareas de integración (servicio de inferencia, pruebas con actuadores o simulación).

En ejecución, trabajo con ciclos diarios de coordinación, donde el equipo comparte avances y obstáculos; el material describe este evento como un “Scrum diario” breve, típicamente con el equipo frente al panel del sprint, para revisar logros, objetivo del día y bloqueos. Aquí, para DomoticAI, los bloqueos típicos suelen ser datos incompletos, latencias, comportamientos inesperados de dispositivos o decisiones de diseño sobre seguridad (por ejemplo, límites de actuación del agente).

Al final del sprint, las salidas principales son el incremento y el feedback formal. El material remarca que el incremento es el resultado de los ítems completados a “terminado” durante el sprint y que, en la revisión, el propietario del producto acepta o rechaza los ítems presentados; si se rechazan, vuelven al product backlog y se re-priorizan. En DomoticAI, un incremento “terminado” podría ser: un pipeline de datos robusto + un modelo entrenado con métricas acordadas + su despliegue en un entorno de prueba con monitorización básica. Tras la revisión, cierro con retrospectiva para definir acciones de mejora del siguiente sprint, coherente con la lógica de inspección y adaptación.

4.3. Roles, artefactos y eventos involucrados

Para operacionalizar Scrum sin repetir definiciones “de manual”, asigno los roles a necesidades concretas del caso. El marco Scrum identifica tres papeles: propietario del producto, equipo de desarrollo y Scrum master; y explica que el propietario del producto maximiza el valor y gestiona la priorización del backlog, el equipo entrega incrementos utilizables, y el Scrum master actúa como líder servicial eliminando impedimentos y entrenando al equipo y a las partes interesadas. En DomoticAI, yo situaría al Product Owner cerca del negocio y del usuario (por ejemplo, alguien responsable de eficiencia energética/confort en el cliente o en la startup), porque debe traducir feedback en prioridades. El equipo de desarrollo lo compongo como un equipo multifuncional (data/ML, integración IoT, backend y calidad), porque el incremento requiere cruzar datos, modelos e integración. El Scrum Master lo enfoco como un facilitador real: asegura disciplina de inspección y adaptación, desbloquea dependencias con fabricantes/instaladores y protege la calidad del proceso.

En artefactos, mantengo los tres mínimos que el material destaca: product backlog, sprint backlog e incremento, porque cubren la trazabilidad de “qué valor quiero” (backlog), “qué me comprometo a entregar ahora” (sprint backlog) y “qué queda realmente disponible y usable” (incremento). El material remarca además que el product backlog se actualiza con feedback de partes interesadas, y que el sprint backlog incluye el trabajo necesario para producir el incremento. En DomoticAI, para que estos artefactos no sean teóricos, anclo cada ítem a una métrica: ahorro estimado, estabilidad, calidad de predicción o reducción de incidencias, de modo que el backlog no sea una lista de tareas sino una lista de hipótesis de valor.

En eventos, opero con la cadencia clásica del sprint: planificación, sincronización diaria, revisión y retrospectiva. El material describe que el sprint arranca con planificación, que existe un Scrum diario para coordinar trabajo y obstáculos, que al final se revisa el resultado y se acepta o rechaza, y que la retrospectiva define acciones para mejorar la fluidez y el resultado del siguiente sprint. En DomoticAI, la revisión la uso para validar con interesados (usuario/operación) que el comportamiento del agente es aceptable, y la retrospectiva para ajustar tanto el proceso (por ejemplo, definición de “terminado” para modelos e integración) como la forma de incorporar feedback sin generar retrabajo.

Referencias utilizadas: [1, 5].

Referencias

- [1] E. Nadal Roig y G. Torruella Fortuny. *La dirección de proyectos: conceptos básicos (PID_00258132)*. Universitat Oberta de Catalunya (UOC).
- [2] P. Mariné Jové y J. R. Rodríguez. *Componentes de la gestión de proyectos: las áreas de conocimiento (PID_00215839)*. Universitat Oberta de Catalunya (UOC).
- [3] A. Gouvea. *Definition (The Crisp-DM Methodology)*. The Crisp-DM Methodology, 2021.
- [4] IBM. *IBM SPSS Modeler CRISP-DM Guide*. IBM, 2011.
- [5] Elena. *Transcripcions en espanyol i català: El método ágil (Manifiesto y principios; Scrum y Kanban)*. Universitat Oberta de Catalunya (UOC), 19 de ago. de 2020.