

# Contents

<b>Capability Distillation Driven Development (CDDD / C3D)</b>	<b>3</b>
Un Whitepaper sobre Navegar la Incertidumbre hacia Capacidades de IA Valiosas . . . . .	3
1. Introducción . . . . .	3
2. Definición de Capability Distillation Driven Development . . . . .	4
3. ¿Qué Significa “Driven” en <i>CDDD / C3D</i> ? . . . . .	4
4. Principios Fundamentales . . . . .	4
5. Intents, Capacidades y Destilación . . . . .	5
5.1 ¿Qué es un Intent? . . . . .	5
5.2 ¿Por Qué “Distillation”? . . . . .	6
5.3 El Proceso de Distillation . . . . .	6
5.4 Branching y el Emergent Roadmap . . . . .	6
6. Fases de Capability Distillation Driven Development . . . . .	7
6.1 Discovery . . . . .	7
6.2 Proof of Concept (PoC) . . . . .	7
6.3 Intent Maturity Checkpoint (IMC) . . . . .	8
6.4 Iterative Refinement . . . . .	9
7. Intent Maturity Threshold (IMT) . . . . .	10
7.1 Naturaleza del IMT . . . . .	10
7.2 Ajustando el IMT . . . . .	10
7.3 Métricas de Ejemplo . . . . .	10
8. Midiendo el Progreso . . . . .	11
8.1 KPI Primario: Distilled Capabilities . . . . .	11
8.2 Gestión de Costos: Costo Promedio por Capability . . . . .	11
8.3 Progression Goals . . . . .	11
8.4 Capability Velocity . . . . .	12
9. Modelo de Evolución . . . . .	12
10. Beneficios del Framework . . . . .	13
10.1 Abordando Desalineación y Mala Comunicación . . . . .	13
10.2 Escapando del “Pilot Purgatory” . . . . .	14
10.3 Previniendo Abandono Post-PoC . . . . .	14
10.4 Cerrando la Brecha Adopción-Impacto . . . . .	14
10.5 Preservando Conocimiento Institucional . . . . .	15
11. Conclusión . . . . .	15
Referencias . . . . .	16
Fuentes Primarias . . . . .	16
Investigación de Soporte . . . . .	16

Análisis de Industria . . . . .	16
Apéndice A: Glosario de CDDD / C3D . . . . .	17
Apéndice B: Glosario Técnico General . . . . .	18
Licencia y Atribución . . . . .	19
Sobre el Autor . . . . .	19

# Capability Distillation Driven Development (CDDD / C3D)

**Versión:** 1.0.2

**Publicado:** 31 de enero de 2026

**Actualizado:** 3 de febrero de 2026

## Un Whitepaper sobre Navegar la Incertidumbre hacia Capacidades de IA Valiosas

---

### 1. Introducción

El desarrollo de capacidades basadas en IA se caracteriza por una alta incertidumbre inicial: no todas las ideas son viables, no todas las intenciones pueden materializarse, y no todas las capacidades entregan suficiente valor para justificar su continuación.

Los enfoques tradicionales de desarrollo de software frecuentemente asumen que el producto final es alcanzable desde el inicio, lo que conduce a planes rígidos, promesas prematuras y esfuerzo desperdiciado en capacidades que nunca llegan a ser verdaderamente útiles.

**Capability Distillation Driven Development (CDDD / C3D)** propone un enfoque alternativo y más honesto: aceptar la incertidumbre como condición inicial y permitir que solo aquellas capacidades que destilen suficiente valor sobrevivan y maduren.

#### La Realidad de la Incertidumbre en Proyectos de IA (2024–2025)

La investigación de la industria muestra consistentemente que el desarrollo de IA conlleva tasas de fracaso significativamente más altas que los proyectos de software tradicionales:

- **Más del 80% de los proyectos de IA fracasan** — el doble de la tasa de fracaso de proyectos de TI sin IA. ([RAND Corporation, 2024](#))
- **El 50% de los proyectos de IA generativa son abandonados** después de la prueba de concepto debido a mala calidad de datos, controles de riesgo inadecuados, costos crecientes o valor de negocio poco claro. ([Gartner, 2025](#))
- **Solo el 39% de las organizaciones reportan algún impacto en EBIT** de iniciativas de IA, a pesar de tasas de adopción del 88%. ([McKinsey Global AI Survey, 2025](#))

Estas cifras subrayan una verdad fundamental: la incertidumbre no es una excepción en el desarrollo de IA — es la norma. Los enfoques que asumen resultados predecibles desde el inicio están desalineados con esta realidad.

---

## 2. Definición de Capability Distillation Driven Development

**Capability Distillation Driven Development (CDDD / C3D)** es un framework de desarrollo de software—especialmente adecuado para capacidades habilitadas por IA—en el cual múltiples intenciones son exploradas y evaluadas progresivamente, aceptando explícitamente que no todas resultarán viables o entregarán suficiente valor.

El desarrollo está *driven* (dirigido) por un proceso continuo de destilación: la incertidumbre se reduce, las capacidades que no alcanzan los umbrales necesarios son eliminadas, y el esfuerzo se concentra únicamente en aquellas que demuestran valor, viabilidad y estabilidad crecientes.

**Nota de alcance:** a lo largo de este whitepaper, “capacidades de IA” se refiere a la capa de integración orientada al negocio alrededor de modelos fundamentales existentes, incluyendo integración con LLMs para tareas específicas o procesamiento de datos, como pipelines RAG, orquestación agéntica, plataformas gestionadas, MLOps/LLMOPs, e integración con sistemas de ciclo de vida y gobernanza empresarial.

---

## 3. ¿Qué Significa “Driven” en *CDDD / C3D*?

En *CDDD / C3D*, el término “driven” indica que las decisiones de desarrollo están **gobernadas por evidencia** obtenida durante el proceso de destilación de capacidades, en lugar de por planes, roadmaps o compromisos de alcance definidos de antemano.

El progreso, aceleración, continuación o discontinuación de una capacidad se determina por **resultados observables** generados en cada fase del proceso.

---

## 4. Principios Fundamentales

1. **No todas las capacidades deben sobrevivir** La discontinuación es un resultado válido y esperado del proceso.
2. **La intención precede a la capacidad** Las capacidades emergen solo si la intención destila valor exitosamente.
3. **La incertidumbre se reduce progresivamente** La certeza no se promete desde el inicio; se construye a través de evidencia.
4. **La inversión sigue a la evidencia** El esfuerzo aumenta solo cuando la capacidad lo justifica.
5. **La velocidad se gana, no se asume** El ritmo de desarrollo aumenta a medida que la incertidumbre disminuye y la intención demuestra viabilidad.

6. **El producto es un residuo concentrado de capacidades valiosas** El producto final es el resultado de múltiples procesos de destilación.
- 

## 5. Intents, Capacidades y Destilación

### 5.1 ¿Qué es un Intent?

En *CDDD / C3D*, un **Intent** representa una hipótesis de capacidad potencial (por ejemplo: clasificar información, extraer conocimiento, asistir decisiones, automatizar análisis).

Un Intent **no es un requerimiento convencional, ni una historia de usuario, ni una característica comprometida**. Es una hipótesis de capacidad que debe ganarse su continuación a través de evidencia. A diferencia de los requerimientos tradicionales—que asumen entregabilidad desde el inicio—un Intent acepta explícitamente que puede no ser viable, puede no entregar suficiente valor, o puede ser descartado en cualquier punto del proceso.

Un Intent puede originarse como una declaración informal de un stakeholder o cualquier miembro del equipo sobre una capacidad deseable. Sin embargo, **un Intent debe ser formalmente definido y documentado** antes de entrar al proceso de destilación. Esta definición formal se captura en el **Intent Journal**—el documento vivo que acompañará al Intent a lo largo del tiempo.

La entrada inicial del Intent Journal debe capturar:

- el alcance completo del Intent,
- su resultado esperado,
- su valor de negocio o impacto anticipado.

Esta formalización asegura que el proceso de destilación permanezca alineado con el propósito de negocio, y previene perseguir capacidades que están desalineadas con los objetivos del sistema o la organización. Sin esta alineación, existe el riesgo de invertir esfuerzo en Intents que son técnicamente interesantes pero estratégicamente irrelevantes.

El Intent Journal debe ser **actualizado como mínimo al final de cada ciclo de destilación**—capturando aprendizajes, decisiones y resultados—para mantener trazabilidad completa a través de toda la línea de tiempo del Intent. Los equipos pueden elegir actualizar el Journal más frecuentemente para mayor visibilidad y control, pero esta granularidad no está prescrita por el framework.

Mantener el Intent Journal rigurosamente proporciona beneficios adicionales:

- **Evitar esfuerzo redundante:** el equipo puede identificar cuando un nuevo Intent duplica o se asemeja cercanamente a un Intent previamente intentado que no fue viable, previniendo inversión desperdiciada en caminos ya explorados.
- **Habilitar re-evaluación:** los Intents fallidos pueden ser revisitados en un momento posterior para evaluar su viabilidad a la luz de avances tecnológicos o circunstancias cambiadas. Lo que no era factible antes puede volverse alcanzable a medida que el panorama evoluciona.
- **Rescatar trabajo reutilizable:** artefactos y piezas de implementación producidos durante Intents pasados—independientemente de si el Intent mismo fue viable—pueden ser identificados y reutilizados en Intents futuros, reduciendo duplicación de esfuerzo.

Además del Intent Journal, **todos los artefactos producidos durante el proceso de destilación deben ser versionados y almacenados históricamente**. Esto permite al equipo revisitar trabajo pasado, recuperar componentes útiles, y construir sobre esfuerzos previos incluso cuando el Intent original no alcanzó completitud.

## 5.2 ¿Por Qué “Distillation”?

La metáfora de **Distillation** (destilación) fue elegida deliberadamente para capturar la naturaleza esencial del desarrollo de capacidades de IA:

- **No todo sobrevive.** Así como la destilación separa la esencia valiosa de la materia prima, *CDDD / C3D* acepta que muchos Intents no resultarán viables—y eso es esperado, no un fracaso.
- **El valor se concentra.** El proceso elimina progresivamente ruido, resultados parciales y caminos de bajo valor, concentrando el esfuerzo en lo que demuestra valor real.
- **El resultado es un residuo refinado.** El producto final no es un entregable predeterminado, sino el resultado acumulado de múltiples ciclos de destilación—una esencia concentrada de capacidades que ganaron su lugar a través de evidencia.

Este encuadre rechaza explícitamente la suposición de que todas las capacidades planificadas serán alcanzadas. En cambio, abraza un proceso selectivo y evolutivo alineado con la incertidumbre inherente del desarrollo de IA.

## 5.3 El Proceso de Distillation

A través del proceso de Distillation:

- algunos Intents son descartados,
- otros se materializan parcialmente,
- y solo unos pocos evolucionan hacia capacidades confiables.

Este proceso es **acumulativo y selectivo**, similar a la destilación técnica: el ruido se elimina, el valor se concentra, y se obtiene una esencia funcional.

## 5.4 Branching y el Emergent Roadmap

Los Intents pueden **ramificarse** en nuevos Intents más especializados una vez que su viabilidad es demostrada. Por ejemplo, un Intent base como “clasificar información” puede habilitar Intents derivados como “extraer entidades”, “validar consistencia”, o “enriquecer con datos externos”.

Al衍生 un nuevo Intent, **el Intent derivado debe ser formalmente definido y documentado** antes de entrar a su propio ciclo de destilación—siguiendo los mismos requerimientos de formalización que cualquier otro Intent. El Intent derivado mantiene una referencia a su padre, preservando trazabilidad a través del Intent Tree.

Este comportamiento de ramificación tiene una implicación fundamental: **el roadmap no es una lista fija de características, sino un árbol (o grafo) emergente de Intents** que se expande a medida que la evidencia lo permite. La planificación en *CDDD / C3D* no se trata de comprometerse con un alcance predefinido, sino de navegar una estructura evolutiva de capacidades validadas y potenciales.

## 6. Fases de Capability Distillation Driven Development

En *CDDD / C3D*, una **fase** no es un estado discreto o un hito que se “alcanza” y se marca como completado. Una fase representa un **período de trabajo continuo** dedicado a un objetivo específico dentro del proceso de Distillation. El trabajo continúa hasta que los criterios de completitud de la fase se cumplen, lo cual entonces habilita la transición a la siguiente fase.

La completitud de fase marca un **hito de viabilidad** para el Intent—no una característica terminada o un release de producción. A diferencia del desarrollo de software tradicional, donde “terminado” frecuentemente significa “desplegado” o “característica completa”, las fases en *CDDD / C3D* señalan confianza creciente en el valor y factibilidad del Intent, sin implicar finalidad.

Esta distinción es esencial: las fases se tratan de **ganarse el derecho a continuar**, no de entregar outputs predeterminados.

### 6.1 Discovery

En el desarrollo de software tradicional, “discovery” es frecuentemente una fase teórica enfocada en investigación, recopilación de requerimientos y evaluación de factibilidad—conducida antes de que se escriba cualquier código. Este enfoque funciona porque el desarrollo de software tradicional opera con certeza significativamente mayor: la viabilidad técnica frecuentemente puede determinarse solo a través de análisis.

En *CDDD / C3D*, Discovery es fundamentalmente diferente. **No es una fase teórica.** Discovery involucra investigación activa combinada con aplicación rápida de esa investigación para evaluar viabilidad. Esto incluye construir artefactos exploratorios, probar hipótesis con código funcional, y validar conocimiento recopilado a través de implementación—todo con el objetivo de establecer un camino que parezca suficientemente viable para justificar un Proof of Concept.

Durante Discovery:

- se prueban pipelines ligeros y flujos exploratorios,
- se aceptan procesos manuales o no reproducibles,
- la prioridad es entender si existe un camino técnico razonable.

Discovery **no busca estabilidad o forma final**, sino evidencia de posibilidad.

Discovery concluye cuando el Intent demuestra suficiente potencial para justificar un Proof of Concept más estructurado, o cuando se toma la decisión de descartar el Intent.

---

### 6.2 Proof of Concept (PoC)

En *CDDD / C3D*, el Proof of Concept **no corresponde al PoC tradicional desecharable** cuyo único propósito es demostrar que algo es posible. Por el contrario, este PoC está explícitamente diseñado para ser **capitalizado como fundamento para desarrollo futuro**.

El PoC representa un punto intermedio crítico en el proceso de Distillation: el momento cuando el Intent ya no es meramente exploratorio pero aún no ha entrado en maduración profunda.

Durante el PoC:

- se consolida una columna vertebral técnica reutilizable,

- se toman decisiones arquitectónicas tempranas pero revisables,
- se valida repetibilidad mínima del flujo,
- se genera evidencia que permite extrapolación confiable de la evolución futura de la capacidad.

El resultado del PoC **no se descarta**. Se preserva como un activo técnico que alimenta directamente las fases subsecuentes de Maturation y Refinement.

El PoC responde la pregunta:

*¿Podemos continuar evolucionando desde aquí con un nivel razonable de certeza?*

La completitud de un PoC indica que el Intent ha alcanzado un nivel suficiente de definición para justificar inversión sostenida en su maduración, no que la capacidad esté completa.

---

### 6.3 Intent Maturity Checkpoint (IMC)

El Intent Maturity Checkpoint (IMC) es la fase dedicada a madurar deliberadamente un Intent hasta que alcance un nivel de confiabilidad, valor y estabilidad definido por un **Intent Maturity Threshold (IMT)**.

Nótese que *CDDD / C3D* deliberadamente evita el término “MVP” (Minimum Viable Product). MVP típicamente implica un hito discreto y un entregable de “mínimo pero ya un producto”. En contraste, el IMC representa un proceso de maduración donde la noción de “producto” es relativa al grado de madurez alcanzado—y donde incluso capacidades maduras pueden re-entrar a fases de Discovery o PoC según la evidencia lo demande.

El IMC constituye el trabajo explícito de capitalizar esfuerzo previo, transformando un Intent validado en PoC en una Distilled Capability que puede:

- sostener inversión futura,
- servir como fundamento estable para Refinement,
- habilitar la derivación de nuevas capacidades o Intents especializados.

Durante la fase de IMC:

- se reduce variabilidad no controlada,
- se fortalecen mecanismos de diagnóstico y control,
- se incrementa el entendimiento operacional del flujo,
- se evalúa sistemáticamente el progreso contra el IMT.

La fase de IMC **no implica cierre definitivo o congelamiento del desarrollo**. Su completitud ocurre cuando el Intent Maturity Threshold es alcanzado, indicando que el Intent ha destilado suficiente valor para ser considerado una capacidad confiable en el contexto actual.

Un solo Intent puede atravesar **múltiples fases de IMC** a lo largo del tiempo, con requerimientos progresivamente más altos. Sin embargo, cada iteración requiere completar el ciclo de vida completo: el Intent debe primero alcanzar su IMT actual, pasar por Refinement, y luego re-entrar a un nuevo ciclo de destilación (Discovery → PoC → IMC) con criterios actualizados. Un Intent no puede iterar dentro de la misma fase de IMC—debe alcanzar su threshold antes de que un nuevo ciclo pueda comenzar.

---

## 6.4 Iterative Refinement

Al alcanzar Refinement, una **Distilled Capability ya existe**—el Intent ha completado exitosamente la fase de IMC y cumplido sus criterios de IMT. Esta capacidad puede proceder hacia producción, integración o entrega independientemente de cualquier decisión de Refinement. Refinement es una oportunidad para mejorar aún más la capacidad, no un prerrequisito para usarla.

Refinement es la fase en la cual una capacidad validada puede continuar siendo destilada para incrementar su pureza, confiabilidad y valor.

A lo largo del proceso de Distillation—desde Discovery hasta IMC—se acumula información valiosa:

- **Aprendizajes:** insights sobre qué funciona, qué no, y por qué.
- **Oportunidades identificadas:** posibilidades que podrían mejorar o extender la capacidad.
- **Deuda técnica:** atajos o compromisos hechos durante exploración rápida que necesitan ser abordados.
- **Derivaciones potenciales:** nuevos Intents que podrían ramificarse de la capacidad actual.

Este conocimiento acumulado sirve como el **input para Refinement**. Un ciclo de Refinement es esencialmente un nuevo paso a través del ciclo de vida completo—Discovery, PoC, IMC—pero informado por evidencia previa y enfocado en mejoras o extensiones específicas.

**No toda Distilled Capability requiere Refinement.** La decisión de refinar debe siempre evaluarse desde la perspectiva de valor de negocio. Una capacidad que entrega resultados aceptables puede no justificar el costo de inversión adicional—incluso si mejoras técnicas son posibles.

Esto es particularmente importante cuando se consideran Refinements de “mejora”: empujar una capacidad funcional hacia niveles de rendimiento más altos. Tales esfuerzos pueden ser desproporcionadamente costosos relativo al impacto de negocio incremental que generan. Antes de iniciar un ciclo de Refinement, es aconsejable:

- Recopilar feedback sobre el **impacto real** de la capacidad como es experimentado por usuarios finales o aplicaciones downstream.
- Evaluar el **valor de negocio potencial** del Refinement propuesto contra su costo potencial.
- Considerar si la inversión estaría mejor dirigida hacia nuevos Intents en lugar de mejorar los existentes.

Un ciclo de Refinement puede enfocarse en:

- eliminar comportamientos inconsistentes,
- mejorar estabilidad y eficiencia,
- abordar deuda técnica,
- reactivar actividades de Discovery o PoC si la evidencia lo demanda.

Estos objetivos se persiguen a través de un nuevo paso del ciclo de vida de Distillation, no como implementación directa dentro de la fase de Refinement misma.

Refinement no es un punto de cierre sino un **punto de continuación potencial**. Basado en la evaluación de conocimiento acumulado y valor de negocio, se toma una decisión sobre si iniciar un nuevo ciclo de destilación ahora, diferirlo para el futuro, o concluir que la capacidad ha alcanzado madurez suficiente para su propósito actual.

Documentar aprendizajes, oportunidades y deuda a lo largo del proceso es esencial para habilitar decisiones de Refinement efectivas—ya sea tomadas inmediatamente o revisitadas después.

---

## 7. Intent Maturity Threshold (IMT)

El **Intent Maturity Threshold (IMT)** es un conjunto de criterios cualitativos y cuantitativos que definen el punto de completitud de un ciclo de trabajo de IMC. El IMT responde la pregunta: *¿qué tan lejos llegaremos en este ciclo?*

El IMT puede establecerse al inicio de la planificación, o puede determinarse o refinarse como resultado de las fases de Discovery y Proof of Concept. Sin embargo, **al menos algunos criterios deben existir**—incluso si inicialmente son cualitativos y definidos vagamente. Sin criterios definidos, el proceso de Distillation carecería de un objetivo claro y podría continuar indefinidamente sin propósito.

Definir el IMT también sirve una función práctica: **limita la inversión** hecha dentro del ciclo de vida del Intent, proporcionando un límite que previene esfuerzo ilimitado en capacidades que pueden no justificar gasto continuo.

### 7.1 Naturaleza del IMT

- **Mixto:** combina métricas cuantitativas y cualitativas.
- **Contextual:** depende del tipo de capacidad y riesgo aceptable.
- **Evolutivo:** puede volverse más estricto a medida que la Distillation progresá a través de ciclos de vida consecutivos, o refinarse basado en evidencia recopilada durante Discovery y PoC.

### 7.2 Ajustando el IMT

Durante la fase de IMC, la evidencia puede indicar que el threshold original debe ajustarse. El framework proporciona guía clara sobre esto:

**Reducir el threshold es aceptable.** Si los resultados observados sugieren que un threshold más bajo aún entrega suficiente valor, el IMT puede reducirse para destilar una capacidad más pronto. Una capacidad que no cumple las expectativas originales puede aún ser suficientemente valiosa para justificar continuación.

**Aumentar el threshold no es recomendado.** Elevar el IMT a mitad de ciclo extiende el alcance de todo el esfuerzo y puede invalidar trabajo realizado en fases previas—trabajo que fue planificado y ejecutado basado en el threshold original. El threshold puede haber sido establecido usando datos de ciclos previos o certeza inicial; aumentarlo socava esa fundación.

Si se necesitan requerimientos más altos, el camino apropiado es completar el ciclo actual con el IMT existente, luego iniciar un nuevo ciclo de destilación con criterios más estrictos a través de Refinement.

### 7.3 Métricas de Ejemplo

#### Cuantitativas (ejemplos):

- tasa de ejecución exitosa,
- latencia promedio y percentiles,

- costo por ejecución,
- cobertura de casos esperados,
- tasa de output aceptable.

#### Cualitativas (ejemplos):

- claridad del flujo de datos,
  - entendimiento de puntos de falla,
  - capacidad de diagnóstico,
  - confianza técnica del equipo,
  - facilidad de evolución del diseño,
  - percepción del usuario final.
- 

## 8. Midiendo el Progreso

*CDDD / C3D* define mecanismos específicos para medir progreso que se alinean con los principios del framework: decisiones basadas en evidencia, inversión acotada, y visibilidad continua.

### 8.1 KPI Primario: Distilled Capabilities

La medida primaria de valor en *CDDD / C3D* es el número de **Distilled Capabilities**—Intents que han pasado exitosamente por IMC y cumplido sus criterios de IMT. Esto es lo que se capitaliza.

Esta métrica puede rastrearse: - **Acumulativamente:** para visualizar resultados a largo plazo y progresión general del proyecto. - **Por período:** para evaluar output reciente y dirección de tendencia.

Las Distilled Capabilities representan funcionalidad validada y lista para producción. Esta es la medida última de si el proceso de Distillation está produciendo valor.

### 8.2 Gestión de Costos: Costo Promedio por Capability

Dada la variabilidad inherente en desarrollo de IA, rastrear el **costo promedio por Distilled Capability** proporciona una línea base práctica para planificación de inversión.

Esta métrica habilita: - **Anticipación de presupuesto:** estimar la inversión requerida para nuevos Intents basado en promedios históricos. - **Decisiones de priorización:** cuando múltiples Intents emergen, contrastar el presupuesto disponible contra el costo promedio ayuda a determinar cuántos pueden perseguirse en un período dado. - **Análisis de tendencia:** rastrear si las capacidades se están volviendo más o menos costosas de destilar a lo largo del tiempo.

### 8.3 Progression Goals

Dentro de cada fase, el trabajo se organiza alrededor de **Progression Goals**—unidades de trabajo a corto plazo, relativamente determinísticas, que son más fáciles de estimar y completar que fases enteras.

Los Progression Goals tienen características específicas: - **Definidos al inicio de fase:** un conjunto inicial de goals se establece cuando una fase comienza. - **Expandidos incrementalmente:** a medida que la fase progresá y más se descubre, goals adicionales se agregan. - **No un alcance fijo:**

el conjunto de goals no está predeterminado para definir duración de fase; más bien, la producción de goals refleja la complejidad evolutiva del trabajo.

Los Progression Goals son la **medida real del rendimiento del equipo**. A diferencia de completitudes de fase o Distilled Capabilities—que dependen de factores más allá del control del equipo—los Progression Goals representan unidades de trabajo concretas y accionables que el equipo entrega directamente.

## 8.4 Capability Velocity

**Capability Velocity** mide el número de Progression Goals completados en un período dado, comparado con el período inmediatamente anterior.

Por ejemplo: - Semana 1: 10 goals completados → línea base - Semana 2: 20 goals completados → velocidad aumentó - Semana 3: 5 goals completados → velocidad disminuyó

Esta métrica: - **Rastrea ritmo del equipo:** proporciona visibilidad sobre cómo el ritmo del equipo está evolucionando. - **Es agnóstica al período:** puede medirse semanal, quincenal o mensualmente dependiendo de las necesidades del equipo. - **No mide eficiencia:** la velocidad refleja ritmo de progresión, no si el trabajo es “bueno” o “eficiente”. - **Es independiente de resultados:** la velocidad se rastrea independientemente de si los Intents ultimadamente resultan en Distilled Capabilities.

Las tendencias de velocidad ayudan a equipos y liderazgo a entender capacidad e identificar cuándo ajustes pueden ser necesarios—sin confundir rendimiento del equipo con la incertidumbre inherente de resultados de capacidades.

---

## 9. Modelo de Evolución

El ciclo de vida de un Intent puede representarse como:

```
Intent Definido (entrada inicial del Intent Journal)
|
v Discovery
|   |-- explorar viabilidad
|   +-+ actualizar Intent Journal
|
v PoC
|   |-- construir fundamento capitalizable
|   +-+ actualizar Intent Journal
|
v IMC (hasta alcanzar IMT)
|   |-- madurar hacia criterios de threshold
|   +-+ actualizar Intent Journal
|
v Refinement
|   |-- Distilled Capability disponible (puede proceder a producción)
|   |-- evaluar: refinrar, derivar, o concluir?
|   +-+ actualizar Intent Journal
```

```

|
| -> [si refinando]
|     |-- extender definición de Intent para alcance de Refinement
|     +-- nuevo ciclo de Distillation (Discovery -> PoC -> IMC -> Refinement)
|
| -> [si derivando]
|     |-- definir nuevo Intent (formalmente documentado)
|     +-- nuevo ciclo de Distillation para Intent derivado
|
+-> [si concluyendo] no más ciclos de Distillation

```

**Importante:** Al alcanzar Refinement, una Distilled Capability ya existe y puede proceder hacia producción independientemente de decisiones de Refinement. (Ver Sección 6.4 para detalles.)

La documentación fluye a lo largo de todo el ciclo de vida: cada fase contribuye aprendizajes, decisiones y artefactos al Intent Journal. En Refinement, el conocimiento acumulado informa si:

- **Refinar:** extender la definición de Intent existente para reflejar el alcance de Refinement, luego ejecutar un nuevo ciclo de Distillation.
- **Derivar:** crear un nuevo Intent formalmente documentado para una capacidad especializada o relacionada, luego ejecutar su propio ciclo de Distillation.
- **Concluir:** determinar que la capacidad ha alcanzado madurez suficiente para su propósito actual.

Este modelo explica:

- por qué el desarrollo no es lineal,
  - por qué el producto final es el resultado de selección progresiva,
  - iteración continua sin hitos absolutos,
  - incremento gradual en velocidad y confianza,
  - el rol central de la documentación en habilitar decisiones informadas a través de ciclos.
- 

## 10. Beneficios del Framework

Los problemas identificados en el desarrollo actual de IA comparten causas raíz comunes que van más allá de estadísticas superficiales. Entender estas causas revela por qué *CDDD / C3D* está diseñado de la manera en que está.

### 10.1 Abordando Desalineación y Mala Comunicación

Según investigación de RAND Corporation, la causa más común de fracaso de proyectos de IA es que los stakeholders malinterpretan o comunican mal qué problema necesita resolverse y qué es capaz de lograr la tecnología. Los equipos técnicos y líderes de negocio frecuentemente operan con suposiciones diferentes, llevando a proyectos que resuelven el problema equivocado o prometen resultados inalcanzables.

*CDDD / C3D* aborda esto a través de:

- **Definición formal de Intent:** cada Intent debe documentarse con su alcance, resultado esperado y valor de negocio antes de entrar al proceso de Distillation—forzando alineación

- entre stakeholders técnicos y de negocio desde el inicio.
- **Validación progresiva:** en lugar de comprometerse con una solución final por adelantado, el proceso valida suposiciones incrementalmente, sacando a la superficie desalineaciones temprano cuando son baratas de corregir.
  - **Decisiones basadas en evidencia:** continuación, aceleración o discontinuación se basa en resultados observables, no en promesas o suposiciones iniciales.

## 10.2 Escapando del “Pilot Purgatory”

La investigación de McKinsey revela que dos tercios de las organizaciones permanecen atrapadas en “pilot purgatory”—un estado donde los proyectos de IA permanecen en experimentación sin alcanzar producción o entregar valor de negocio. Esto sucede cuando los pilotos son tratados como experimentos aislados en lugar de pasos hacia integración real.

*CDDD / C3D* está explícitamente diseñado para prevenir esto:

- **PoC es capitalizable, no desecharable:** el Proof of Concept construye un fundamento técnico reutilizable que alimenta directamente las fases subsecuentes, asegurando que el trabajo de piloto nunca se descarte.
- **Criterios claros de progresión:** el Intent Maturity Threshold define criterios explícitos para completitud de fase, previniendo experimentación indefinida.
- **Listo para producción al completar IMC:** al alcanzar Refinement, una Distilled Capability ya existe y puede proceder a producción inmediatamente—Refinement es opcional, no un prerequisito para despliegue.

## 10.3 Previniendo Abandono Post-PoC

Gartner reporta que el 50% de los proyectos de IA generativa son abandonados después del proof of concept debido a mala calidad de datos, controles de riesgo inadecuados, costos crecientes o valor de negocio poco claro. Estos no son fracasos tecnológicos—son fracasos de proceso que ocurren cuando los proyectos carecen de estructura para gestionar incertidumbre.

*CDDD / C3D* mitiga cada una de estas causas:

- **Mala calidad de datos:** la fase de Discovery explícitamente prueba viabilidad de datos antes de inversión significativa; los problemas salen a la superficie temprano en lugar de después de completar el PoC.
- **Costos crecientes:** los criterios de IMT acotan la inversión dentro de cada ciclo, previniendo gasto descontrolado. La decisión de continuar siempre se sopesa contra costo potencial y valor de negocio.
- **Valor de negocio poco claro:** la definición formal de Intent requiere documentar impacto de negocio esperado por adelantado; las decisiones de Refinement explícitamente evalúan impacto real y valor potencial contra costo potencial.
- **Controles de riesgo inadecuados:** el enfoque por fases con checkpoints explícitos proporciona puntos de control naturales para evaluación de riesgo y corrección de curso.

## 10.4 Cerrando la Brecha Adopción-Impacto

A pesar de tasas de adopción de IA del 88%, solo el 39% de las organizaciones reportan algún impacto en EBIT, y solo el 6% logra resultados transformadores. La investigación indica que esta

brecha proviene de hesitación del liderazgo, pobre coordinación estratégica, y tratar la IA como proyectos de tecnología en lugar de iniciativas de transformación de negocio.

*CDDD / C3D* aborda las causas estructurales de esta brecha:

- **Alineación de negocio es obligatoria:** los Intents no pueden entrar al proceso de Distillation sin valor de negocio documentado, asegurando que cada esfuerzo esté atado a resultados estratégicos.
- **La inversión sigue a la evidencia:** los recursos aumentan solo a medida que las capacidades prueban su valor, previniendo grandes compromisos iniciales sobre suposiciones no validadas.
- **Estructura lista para transformación:** el Emergent Roadmap (Intent Tree) naturalmente soporta iniciativas ambiciosas y transformadoras al permitir que las capacidades se ramifiquen y compongan a medida que la evidencia se acumula.

## 10.5 Preservando Conocimiento Institucional

Una causa frecuentemente pasada por alto del fracaso de proyectos de IA es la pérdida de conocimiento institucional—los equipos repiten errores, duplican esfuerzos en enfoques previamente intentados, o abandonan trabajo que podría ser valioso en circunstancias cambiadas.

*CDDD / C3D* trata el Intent Journal como una preocupación de primera clase:

- **Trazabilidad completa del ciclo de vida:** el Intent Journal se actualiza como mínimo al final de cada ciclo de Distillation, preservando la historia completa de decisiones y resultados.
  - **Preservación de artefactos:** todos los artefactos son versionados y almacenados históricamente, habilitando reutilización a través de Intents.
  - **Capacidad de re-evaluación:** documentación rigurosa en el Intent Journal permite que Intents fallidos sean revisitados cuando la tecnología avanza o las circunstancias cambian, transformando “fracasos” pasados en oportunidades futuras.
- 

## 11. Conclusión

Estamos en un punto de inflexión en la evolución del desarrollo de software. La emergencia de capacidades de IA ha introducido un nivel de incertidumbre que las metodologías tradicionales nunca fueron diseñadas para manejar. Sin embargo, las organizaciones continúan aplicando esas mismas metodologías—planificación rígida, compromisos por adelantado, ejecución lineal—esperando resultados diferentes. Las estadísticas citadas a lo largo de este whitepaper no son evidencia de fracaso tecnológico; son evidencia de desajuste metodológico.

El avance que *CDDD / C3D* representa es el reconocimiento de que **el desarrollo de capacidades de IA requiere un enfoque fundamentalmente diferente**—uno que acepta la incertidumbre como condición inicial en lugar de tratarla como un problema a eliminar a través de mejor planificación.

Este cambio tiene implicaciones profundas para cómo las organizaciones invierten sus recursos. Hoy, mucho del esfuerzo vertido en iniciativas de IA está impulsado por presión competitiva—un miedo de quedarse atrás en lugar de un entendimiento claro de creación de valor. El resultado es predecible: experimentos dispersos, pilotos abandonados, e inversiones que generan actividad sin impacto.

*CDDD / C3D* ofrece un camino hacia **transformación de IA responsable**—donde la inversión sigue a la evidencia, donde la discontinuación es un resultado válido en lugar de un fracaso, y donde cada esfuerzo está explícitamente atado a valor de negocio. Esto no se trata de desacelerar; se trata de dirigir energía hacia resultados que importan.

El framework empodera a las organizaciones para tomar propiedad de su proceso de transformación de IA: para tomar decisiones informadas sobre dónde invertir, cuándo continuar, y cuándo parar. Transforma el desarrollo de IA de un impulso por mantenerse competitivo en un esfuerzo coordinado que produce crecimiento de negocio real y medible.

La oportunidad ante nosotros es significativa. Las organizaciones que adopten este framework no solo reducirán esfuerzo desperdiciado—construirán capacidad institucional para navegar incertidumbre sistemáticamente, acumulando conocimiento y artefactos que se componen a lo largo del tiempo. Aquellas que continúen con enfoques tradicionales continuarán experimentando las tasas de fracaso que la investigación actual documenta.

*CDDD / C3D* no es meramente una metodología. Es un framework—un reconocimiento de que las reglas han cambiado, y que el éxito en desarrollo de IA requiere el coraje de cambiar con ellas.

---

## Referencias

### Fuentes Primarias

1. **RAND Corporation (2024).** *The Root Causes of Failure for Artificial Intelligence Projects and How They Can Succeed: Avoiding the Anti-Patterns of AI.* [https://www.rand.org/pubs/research\\_reports/RRA2680-1.html](https://www.rand.org/pubs/research_reports/RRA2680-1.html)
2. **Gartner (2025).** *Why Half of GenAI Projects Fail.* <https://www.gartner.com/en/articles/genai-project-failure>
3. **McKinsey & Company (2025).** *The State of AI: Global Survey 2025.* <https://www.mckinsey.com/capabilities/insights/the-state-of-ai>

### Investigación de Soporte

4. **S&P Global Market Intelligence (2025).** Investigación sobre tasas de abandono de iniciativas de IA (42% en 2025, arriba del 17% en 2024).
5. **MIT (2025).** *The GenAI Divide: State of AI in Business 2025.* Investigación sobre pilotos de IA empresarial y ROI.
6. **Gartner (2024).** Investigación sobre tasas de piloto-a-producción de IA (48% de pilotos de IA alcanzan producción).

### Análisis de Industria

7. **Infosys (2025).** *AI Adoption vs Impact: Closing the Gap for Real Business Value.* <https://blogs.infosys.com/emerging-technology-solutions/iedps/ai-adoption-vs-impact-closing-the-gap-for-real-business-value.html>

8. **Astrafy (2025).** *Scaling AI from Pilot Purgatory: Why Only 33% Reach Production and How to Beat the Odds.* <https://astrafy.io/the-hub/blog/technical/scaling-ai-from-pilot-purgatory-why-only-33-reach-production-and-how-to-beat-the-odds>
  9. **Process Excellence Network (2025).** *Data Foundations for Scalable AI.* (Datos de Gartner sobre mala calidad de datos costando \$12.9M anualmente). <https://www.processexcellencenetwork.com/ai/article/strong-data-foundations-are-the-real-prerequisite-for-scalable-ai>
  10. **Netguru (2025).** *Why Most AI POCs Fail — and How Leaders Can Fix It.* <https://www.netguru.com/blog/why-most-ai-pocs-fail>
  11. **Devoteam (2025).** *Why AI Projects Fail.* <https://www.devoteam.com/expert-view/why-ai-projects-fail/>
- 

*Referencias accedidas en enero 2026. Estadísticas y hallazgos reflejan investigación publicada entre 2024–2025.*

---

## Apéndice A: Glosario de CDDD / C3D

Los siguientes términos tienen significados específicos dentro del framework *Capability Distillation Driven Development*:

**Intent** — Una hipótesis de capacidad potencial que debe ganarse su continuación a través de evidencia. A diferencia de requerimientos tradicionales o historias de usuario, un Intent acepta explícitamente que puede no ser viable y puede ser descartado en cualquier punto.

**Distillation** — El proceso selectivo a través del cual los Intents son progresivamente validados, y solo aquellos que demuestran suficiente valor, viabilidad y estabilidad sobreviven. La metáfora captura que no todo sobrevive, el valor se concentra, y el output es un residuo refinado.

**Discovery** — La fase inicial de exploración activa donde la viabilidad se prueba a través de construcción y experimentación—no una fase de investigación teórica. Incluye crear artefactos exploratorios y validar hipótesis con código funcional.

**Proof of Concept (PoC)** — En *CDDD / C3D*, un fundamento capitalizable para desarrollo futuro—no una demostración desecharable. El PoC construye una columna vertebral técnica reutilizable que alimenta directamente las fases subsecuentes.

**Intent Maturity Checkpoint (IMC)** — La fase dedicada a madurar deliberadamente un Intent hasta que alcance criterios definidos. Es un período de trabajo continuo, no un evento de evaluación discreto.

**Intent Maturity Threshold (IMT)** — El conjunto de criterios cualitativos y cuantitativos que definen el punto de completitud de un ciclo de trabajo de IMC. Responde la pregunta: *¿qué tan lejos llegaremos en este ciclo?*

**Refinement** — Un punto de continuación potencial donde se toma la decisión de extender, derivar o concluir la capacidad. No es un prerequisito para producción—las Distilled Capabilities pueden desplegarse inmediatamente.

**Distilled Capability** — Una capacidad que ha pasado exitosamente a través del proceso de Distillation y cumplido sus criterios de IMT. Representa funcionalidad validada y lista para producción.

**Intent Tree / Emergent Roadmap** — La estructura no lineal de Intents que se expande a medida que la evidencia lo permite. Los Intents pueden ramificarse en Intents derivados, formando un árbol o grafo en lugar de una lista fija de características.

**Intent Journal** — El documento vivo que captura la definición formal de un Intent y acumula entradas a lo largo del tiempo. Incluye aprendizajes, decisiones y resultados de cada ciclo de Distillation. Como mínimo, actualizado al final de cada ciclo de Distillation; los equipos pueden actualizar más frecuentemente para mayor visibilidad.

**Progression Goals** — Unidades de trabajo a corto plazo, relativamente determinísticas, definidas dentro de cada fase. Establecidos al inicio de fase y expandidos incrementalmente a medida que el trabajo progresá. La medida real del rendimiento del equipo, independiente de si los Intents resultan en Distilled Capabilities.

**Capability Velocity** — El número de Progression Goals completados en un período dado, comparado con el período inmediatamente anterior. Mide ritmo del equipo y paso de progresión, no eficiencia o calidad de resultado.

---

## Apéndice B: Glosario Técnico General

Los siguientes términos se usan a lo largo de este documento y pueden ser útiles para lectores menos familiarizados con terminología de desarrollo de software:

**Artefacto** — Cualquier output tangible producido durante desarrollo: código, documentación, configuraciones, modelos, datasets, u otros entregables.

**Pipeline** — Una secuencia de pasos automatizados que procesan datos o código, típicamente incluyendo etapas como ingestión de datos, transformación, validación y generación de output.

**Deuda Técnica** — Atajos o compromisos hechos durante desarrollo que requerirán esfuerzo futuro para abordar. La deuda técnica acumulada puede desacelerar desarrollo futuro y reducir confiabilidad del sistema.

**EBIT (Earnings Before Interest and Taxes)** — Una métrica financiera que mide la rentabilidad operacional de una empresa, excluyendo costos de financiamiento e impuestos. Usado para evaluar si las inversiones en IA generan retornos de negocio reales.

**Proof of Concept (PoC) — Definición Tradicional** — En desarrollo de software convencional, una demostración desecharable de que algo es técnicamente posible, típicamente descartada después de validación. (Comparar con la definición de *CDDD / C3D* arriba.)

**Producción** — El ambiente en vivo donde el software sirve a usuarios reales o procesos de negocio, en oposición a ambientes de desarrollo o pruebas.

**Piloto / Pilot Purgatory** — Un piloto es un despliegue limitado para probar viabilidad. “Pilot purgatory” se refiere al estado donde los proyectos permanecen atascados en experimentación sin alcanzar producción o entregar valor de negocio.

---

## Licencia y Atribución

© 2026 **Manuel Antonio Lara Pupo**. Todos los derechos reservados. Publicado por **Unlimitech Cloud LLC**.

Este trabajo está licenciado bajo la **Licencia Creative Commons Atribución-CompartirIgual 4.0 Internacional (CC BY-SA 4.0)**.

Eres libre de compartir y adaptar este material para cualquier propósito, incluyendo uso comercial, siempre que: - Se dé crédito apropiado al autor original - Las adaptaciones se distribuyan bajo la misma licencia

Licencia completa: <https://creativecommons.org/licenses/by-sa/4.0/>

### Cómo citar:

Lara Pupo, Manuel Antonio. *Capability Distillation Driven Development (CDDD / C3D): Un Whitepaper sobre Navegar la Incertidumbre hacia Capacidades de IA Valiosas*. Versión 1.0.2. Unlimitech Cloud LLC, 31 de enero de 2026.

## Sobre el Autor

**Manuel Antonio Lara Pupo** es un Arquitecto de Soluciones impulsado por IA con más de 20 años de experiencia construyendo software a escala y más de 11 años diseñando arquitectura empresarial. Se especializa en traducir objetivos de negocio complejos en soluciones robustas y seguras—incluyendo plataformas a escala nacional. Su trabajo reciente se enfoca en habilitación de IA en producción: flujos de trabajo agénticos, implementaciones MCP y pipelines de datos inteligentes. Manuel es el fundador de Unlimitech Cloud LLC, donde combina profunda experiencia técnica con el compromiso de compartir conocimiento que fortalece la comunidad de ingeniería.

---

### Nota de Transparencia

Este whitepaper es una obra intelectual original del autor. El framework, conceptos y metodología presentados aquí fueron desarrollados basándose en experiencia profesional, observación directa y casos de uso reales.

Las herramientas de IA se utilizaron para asistir con investigación, análisis de grandes conjuntos de datos, redacción, edición y refinamiento del lenguaje. Todo el contenido fue revisado y verificado por el autor.

---

Fuente Original: [Capability Distillation Driven Development-v1.0.2-ES-signed.pdf](#)