

THE FUTURE OF WORK · LIBRO I

MÁS ALLÁ DE LA NÓMINA

*Cómo dirigir una empresa
cuando el mejor talento
no está en nómina*

IGOR FERNÁNDEZ



THE FUTURE OF WORK · LIBRO I · EDICIÓN 2.0

MÁS ALLÁ DE LA NÓMINA

*Cómo dirigir una empresa cuando el mejor talento
no está en nómina*

IGOR FERNÁNDEZ

The Future of Work Series — Libro I.

Edición final · versión 2.0 · 2026.

© 2026 Igor Fernández. Todos los derechos reservados. Ninguna parte de esta publicación puede reproducirse, distribuirse o transmitirse por ningún medio sin autorización previa por escrito del titular de los derechos.

Impreso en España.

Nota del autor

Este libro parte de una incomodidad práctica. Las empresas siguen planificando buena parte de su capacidad a través de puestos y headcount, aunque sus resultados dependen también de especialistas independientes, firmas asociadas, proveedores, comunidades, software y sistemas de inteligencia artificial. Esa diferencia entre cómo contamos la organización y cómo trabaja de verdad no convierte el empleo en un error. Convierte el organigrama en un mapa incompleto.

La investigación que sigue no busca demostrar que el futuro pertenezca a empresas sin empleados. Busca responder una pregunta más útil: **¿cómo puede una organización acceder a capacidades que no necesita —o no puede— mantener permanentemente en nómina, sin confundir acceso con disponibilidad ni flexibilidad con transferencia de riesgo?**

Cuando el texto formula un framework propio, lo presenta como una herramienta que debe probarse. Cuando una idea procede de literatura anterior, reconoce su genealogía. Cuando una afirmación es una predicción, aparece como tal. Los ejemplos empresariales ilustran mecanismos y límites; ninguno demuestra una ley universal.

CONTENIDO

Índice

Prefacio · Nota del autor

Introducción · La empresa que el organigrama no ve

Parte I · El valor por defecto de la nómina

- 1 The Payroll Company
- 2 The Talent Boundary
- 3 Controlled, Committed, Accessible

Parte II · Más allá de la contratación

- 4 Build / Buy / Borrow / Automate
- 5 Fractional Everything?
- 6 The Unbundling and Rebundling of the Job

Parte III · La fuerza de trabajo híbrida

- 7 Workforce as a Portfolio
- 8 What Must Stay Close?
- 9 The New Talent Architecture

Parte IV · De la captación a la orquestación del talento

- 10 The Talent Cloud
- 11 Talent Orchestration
- 12 Leading Without Employment Authority

Parte V · La organización permeable

- 13 Designing for Access
- 14 The Permeable Company

Material final

- [Glosario](#) · [Herramientas](#) · [Índice de figuras](#) · [Bibliografía](#)

La empresa que el organigrama no ve

Una empresa no coincide con la lista de personas que cobran una nómina. Nunca coincidió del todo. Una fábrica dependía de proveedores, distribuidores, asesores, energía e infraestructuras públicas mucho antes de internet. Una firma profesional recurría a expertos, corresponsales y tecnología. Un comercio movilizaba fabricantes, logística, pagos y conocimiento que no poseía.

Lo que cambia no es la existencia de lo externo, sino la variedad y la velocidad con que ciertas capacidades pueden conectarse. Trabajo remoto, cloud, software como servicio, marketplaces, redes profesionales, pagos internacionales, comunidades abiertas e inteligencia artificial amplían el menú. Una empresa pequeña puede utilizar infraestructura y expertise que antes exigían escala. Una organización grande puede formar configuraciones alrededor de un problema sin incorporar permanentemente cada skill.

Ese menú crea una ilusión. Ver más opciones puede parecer equivalente a disponer de más capacidad. No lo es. Un perfil visible no está reservado. Un proveedor contratado no conoce automáticamente el contexto. Una herramienta que produce una respuesta no posee el workflow. Una comunidad no acepta los plazos de la empresa. Una idea obtenida fuera todavía debe integrarse. El acceso abre una posibilidad; la organización produce la capacidad.

Este libro estudia el trabajo que hay entre ambos estados.

Una tesis que sobrevivió a su versión más atractiva

La formulación maximalista sería sencilla: el talento ya no necesita pertenecer a la empresa; poseer será sustituido por acceder; las organizaciones del futuro tendrán muy pocos empleados y una fuerza laboral prácticamente infinita. Es memorable y no está demostrada.

El empleo sigue resolviendo continuidad, autoridad, aprendizaje, inversión, identidad, protección y team production. Los mercados mantienen fricciones. La automatización necesita personas, datos, capital y assurance. Una estructura ligera para la empresa puede descansar sobre trabajo y riesgo desplazados. Las fronteras no desaparecen; se multiplican y, a menudo, se vuelven menos visibles.

La tesis que resiste es más útil:

Una organización puede movilizar capacidades más allá de su plantilla cuando elige una modalidad que encaja con el trabajo, diseña interfaces suficientes, aprende a orquestar y conserva control estable sobre aquello de lo que necesita seguir respondiendo.

Cada verbo limita al anterior. **Movilizar** exige más que descubrir. **Elegir** excluye una modalidad universal. **Diseñar** reconoce el coste de integración. **Aprender** exige memoria. **Conservar control** impide que apertura signifique vacío. **Responder** mantiene la dimensión institucional.

El resultado no es necesariamente una empresa con menos empleo. Puede contratar más donde el conocimiento y la continuidad lo justifican, abrir otras capacidades y automatizar pasos. La transformación ocurre en la calidad de la elección.

De personas a capacidades, sin deshumanizar

La unidad de análisis del libro es la **capability**: facultad de un sistema para producir un outcome de forma repetible bajo umbrales de calidad, tiempo, coste, aprendizaje y recuperación. Puede combinar personas empleadas, independientes, proveedores, software, IA, datos y activos. Ninguno de esos componentes es equivalente a los demás.

Utilizar capability evita dos errores. El primero es tratar una skill, una persona o una licencia como si ya fuera un resultado. El segundo es diseñar a partir del contrato. Si la necesidad se expresa como puesto, el empleo gana por definición; si se expresa como factura, gana el proveedor; si se expresa como feature, gana la herramienta. Outcome y capability permiten comparar.

La palabra también presenta un peligro: convertir a las personas en unidades fungibles. El libro lo evita manteniendo dos balances. La empresa necesita performance. Quienes contribuyen mantienen derechos, autonomía, carrera, dependencia y exposición al riesgo. Una arquitectura no es mejor si mejora un balance ocultando el otro.

Por eso se abandona la expresión *owned talent* salvo como crítica histórica. Las empresas no poseen personas. Pueden emplear, reservar capacidad, controlar activos, asignar decision rights y acceder a opciones. El lenguaje debe reflejar la relación real.

Las cinco preguntas

La progresión del libro responde cinco preguntas.

La primera es de **visibilidad**: ¿qué trabajo, capital y dependencias producen realmente el outcome? El Capability Baseline corrige el headcount como denominador universal.

La segunda es de **frontera**: ¿qué fuente está solo visible y cuál puede producir de manera fiable? El Capability Usability Gate separa catálogo de capacidad efectiva.

La tercera es de **elección**: ¿qué desarrollar, incorporar, acceder o automatizar? BBBA, Fractional Fit y Separability Test convierten la modalidad en hipótesis reversible.

La cuarta es de **operación**: ¿cómo se conecta una portfolio plural? Access Layers, Interface Contract y Orchestration Loop hacen visible integración, assurance, memoria y carga directiva.

La quinta es de **institución**: ¿qué debe permanecer y qué puede cruzar? Solid Minimum y Permeability Profile permiten abrir y cerrar de forma selectiva.

Las preguntas no forman una escalera hacia más externalización. Una respuesta puede conducir a empleo, otra a una relación committed, otra a mercado y otra a automatización. El destino es un diseño explicable.

Evidencia, casos y límites

Los casos cumplen funciones delimitadas. NASA y P&G muestran mecanismos de acceso; GitLab, documentación distribuida; Red Hat y Linux, producción abierta con institución; Apple y aviación, control y dependencia en redes; plataformas de talento, mercados e intermediación. Ninguno demuestra una forma universal.

Las fuentes corporativas permiten describir prácticas declaradas, no probar su efecto. Los datos laborales dependen de definiciones: *independent contractor* en el empleo principal no equivale a todo trabajo externo. Los estudios de IA miden tareas y contextos fechados, no ocupaciones completas ni autonomía empresarial.

Cuando la evidencia no permite prevalencia o causalidad, el libro formula hipótesis y pruebas. *Fractional Everything* conserva un signo de interrogación. Talent Orchestration se presenta como operacionalización que debe demostrar valor incremental. Los maturity profiles no suman una nota. Las afirmaciones jurídicas se limitan a su jurisdicción y requieren revisión profesional.

La cautela no debilita el argumento. Le permite distinguir lo que ya sabemos de lo que solo resulta plausible.

Una guía para leer

Cada capítulo puede leerse como una decisión. Si dirige una empresa pequeña, aplique los frameworks a una capacidad que hoy depende demasiado de usted. Si trabaja en una organización grande, elija un outcome que atravesase People, Procurement y Tecnología. No intente mapearlo todo.

Registre la hipótesis antes de actuar. ¿Por qué espera que la modalidad elegida mejore velocidad, calidad, aprendizaje o resiliencia? ¿Qué alternativa descarta? ¿Quién absorbe riesgo? ¿Qué señal le haría cambiar? Estas preguntas convierten el libro en un experimento y protegen contra el uso ceremonial de sus nombres.

Los términos ingleses permanecen cuando forman parte de la arquitectura o pierden precisión al traducirse. No deben utilizarse para ocultar una idea simple. *Outcome* señala un estado verificable, no un entregable. *Assurance* reúne evaluación y control antes del daño. *Fallback* describe una ruta de recuperación. *Solid Minimum* expresa la tensión entre apertura y estabilidad.

La promesa

Al final, el lector debería poder tomar una solicitud de contratación, un contrato o una propuesta de automatización y reconstruir la pregunta anterior. Debería ver qué parte del sistema está oculta, qué modalidad encaja, qué interfaz falta, qué conocimiento debe permanecer y quién responde.

Si la conclusión es contratar, el framework habrá funcionado. Si es acceder a una red, también. Si es no automatizar todavía, habrá producido valor. *Beyond Payroll* no mide progreso por distancia respecto a la nómina. Lo mide por distancia respecto al default.

La empresa que el organigrama no ve ya existe. El desafío es aprender a gobernarla sin perder aquello que hizo útil a la empresa como institución.

Qué cambia el lunes por la mañana

El argumento puede parecer una teoría amplia sobre el futuro del trabajo. Su prueba comienza en decisiones ordinarias. Un manager pide una vacante porque el equipo no llega. Procurement recibe una propuesta de una agencia. Tecnología quiere conectar una herramienta. Finanzas pregunta por qué crece el gasto externo. People intenta clasificar una relación que ya opera como parte del equipo. Cada función ve un objeto distinto y ninguna ve el outcome completo.

La primera práctica consiste en detener la traducción automática. La solicitud de vacante vuelve a expresarse como capacidad; la propuesta del proveedor se dibuja dentro del workflow; la herramienta se evalúa con preparación y review; el gasto se reconcilia con trabajo y resultado; la relación se describe por conducta, dependencia y derechos. Este paso puede confirmar la decisión original. Su valor está en hacerla comparable.

La segunda práctica es nombrar al owner end-to-end. No a la persona que procesa el contrato ni a quien administra la herramienta, sino a quien puede aceptar el outcome y responder cuando las piezas discrepan. Si no existe, la organización ha encontrado un problema anterior al sourcing.

La tercera es registrar la hipótesis. “Contratamos porque necesitamos continuidad y aprendizaje”; “usamos una firma porque la demanda es intermitente y el mercado posee expertise”; “automatizamos porque el paso es evaluable y repetitivo”. La frase incluye una señal de error. Sin ella, la modalidad se convierte en identidad y cada dato se interpreta para defenderla.

La cuarta es mirar el otro balance. ¿Qué trabajo adicional recae en el núcleo? ¿Qué disponibilidad se espera de la contraparte? ¿Qué coste, riesgo o aprendizaje cambia de manos? La arquitectura puede seguir adelante después de observarlo. La diferencia es que deja de llamar desaparición a una transferencia.

La quinta es diseñar la salida antes de celebrar la entrada. Datos, permisos, artefactos, memoria, pagos y continuidad deben tener una ruta. Este ejercicio suele mejorar también el onboarding: obliga a saber qué objeto se está conectando.

Estas prácticas no requieren un programa de transformación. Pueden aplicarse a una sola decisión material y producir evidencia en semanas. Si mejoran claridad, ramp, calidad o recovery, se amplían. Si añaden burocracia sin cambiar outcome, se simplifican. El propio libro queda sometido a su criterio de contingencia.

La frontera como conversación común

El cambio más profundo quizá sea lingüístico. Durante años, People,

Procurement, Operaciones y Tecnología han utilizado unidades distintas: puesto, contrato, proceso y sistema. Cada una es legítima y cada una pierde una parte. Capability ofrece un punto de encuentro sin borrar especialidades. Permite discutir el mismo outcome y después devolver a cada función sus responsabilidades específicas.

Esa conversación reduce el arbitraje interno. Un equipo no puede llamar proveedor a una relación que gestiona como empleo para evitar a People. Tampoco puede llamar transformación tecnológica a una automatización cuyo review recae en Operaciones. Finanzas puede pedir el coste total; Riesgo, el camino de recuperación; quienes trabajan, claridad sobre la relación y la evaluación.

El lenguaje común no garantiza acuerdo. Hace visibles los desacuerdos correctos: cuánto control merece una capacidad, qué riesgo aceptar, dónde aprender y qué flexibilidad financiar. Una organización no se vuelve abierta porque adopta términos nuevos. Se vuelve más deliberada cuando esos términos obligan a explicar decisiones antes ocultas por la categoría.

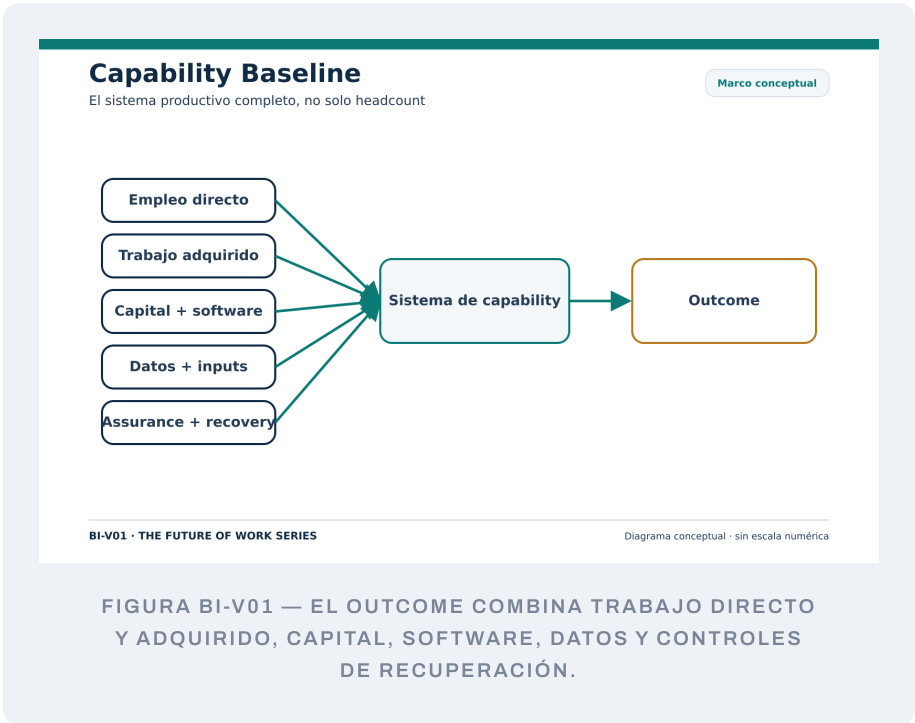
PARTE I

EL VALOR POR DEFECTO DE LA NÓMINA

La nómina es al mismo tiempo una institución y una lente. Como institución reúne continuidad, autoridad, aprendizaje, protección y capacidad de inversión. Como lente deja fuera una parte creciente del sistema que produce los resultados de una empresa. Esta primera parte no propone abandonar payroll. Propone aprender a ver lo que payroll no cuenta.

Capítulo 1

THE PAYROLL COMPANY



La empresa que aprendimos a contar

Abra el informe anual de una compañía grande y busque el número de empleados. La cifra suele estar ahí, sobria y precisa. Ahora intente encontrar otra: cuántas horas de trabajo humano, cuántos proveedores críticos, cuánto software, qué comunidades y qué infraestructura fueron necesarios para producir todo lo que la compañía vendió ese año. La segunda cifra casi nunca existe en una sola línea.

No es una omisión accidental. El headcount responde a preguntas legítimas: ¿cuántas relaciones laborales directas sostiene la empresa?, ¿qué coste fijo aproxima?, ¿qué obligaciones tiene?, ¿cómo está distribuida su plantilla? Pero hemos permitido que responda también preguntas para las que no fue diseñado: ¿cuánta capacidad tiene la compañía?, ¿cuánto trabajo contiene su producto?, ¿es más productiva que otra?, ¿qué necesita para crecer?

Piense en Apple. La empresa publica información sobre sus empleados y, por separado, describe una cadena de suministro que comprende fabricación, ensamblaje, materiales, logística y proveedores distribuidos por numerosos países.¹ Ninguno de los dos mapas es falso. El problema aparece cuando tratamos el primero como si contuviera al segundo. Una persona que mira únicamente el organigrama ve una compañía. Una persona que sigue el recorrido de un dispositivo ve un sistema productivo.

La misma diferencia existe en una firma pequeña. Un negocio registrado sin payroll puede descansar sobre el trabajo de su fundador, servicios profesionales, una plataforma de comercio, cloud, logística y capital. Las **Nonemployer Statistics** del Census Bureau cuentan negocios estadounidenses sin empleados remunerados, no negocios sin trabajo.² El matiz parece obvio. Sin embargo, gran parte de la conversación sobre empresas “de una persona” o “sin empleados” lo pierde por el camino.

Por qué payroll se convirtió en la empresa

Durante la industrialización, reunir personas y activos dentro de una organización estable resolvía problemas que el mercado no resolvía con la misma eficiencia. Ronald Coase formuló la pregunta decisiva en 1937: si los mercados coordinan la actividad mediante precios, ¿por qué existen las firmas? Su respuesta abrió una

tradición: usar el mercado también cuesta. Hay que encontrar a la contraparte, negociar, verificar, adaptar y volver a contratar.³

La nómina no fue simplemente un método de pago. Formó parte de un bundle institucional. Una relación laboral duradera ofrecía disponibilidad, un marco de autoridad, identidad compartida, aprendizaje acumulado y un modo de absorber variaciones de demanda. Permitía que la empresa invirtiera en habilidades cuyo retorno llegaría más tarde. Hacía posible coordinar team production cuando el resultado no podía atribuirse limpiamente a una sola contribución. También trasladaba una parte del riesgo económico desde la persona hacia la institución: el salario no tenía que renegociarse para cada tarea.

La gran empresa industrial añadió capital, departamentos, procedimientos y una jerarquía profesional. Alfred Chandler mostró cómo la coordinación administrativa llegó a superar al mercado en actividades donde el volumen, la velocidad y la interdependencia hacían valiosa una dirección continua.⁴ El empleo se volvió la forma visible de acumular capacidad porque resolvía, en un mismo contrato social y organizativo, muchos problemas a la vez.

Ese bundle sigue siendo extraordinariamente eficiente en contextos que requieren continuidad, conocimiento tácito, seguridad, aprendizaje conjunto o respuesta inmediata. La empresa que contrata a una persona no compra solamente horas. Construye una relación con derechos, expectativas, inversión y memoria. Por eso la tesis de este libro no es “emplear menos”. Es “dejar de asumir que toda capacidad necesaria debe tomar la forma de un puesto permanente”.

La cifra que tranquiliza

El headcount tiene otra ventaja: es comprensible. Un comité de dirección puede discutir diez contrataciones con más facilidad que un sistema de capacidades repartido entre empleados, especialistas, agencias, proveedores y software. La plantilla encaja en presupuesto, organigrama y ownership. Cada puesto tiene una caja y cada caja, idealmente, un manager.

La claridad tiene un precio. Al convertir la caja en unidad de planificación, tendemos a hacer tres suposiciones.

La primera es que una necesidad empresarial equivale a un puesto. “Necesitamos mejorar la fijación de precios” se convierte en “necesitamos contratar a un pricing manager”. Hemos saltado del outcome a la modalidad sin preguntar si la necesidad es continua, divisible, estratégica o automatizable.

La segunda es que lo que no está dentro resulta secundario. Sin embargo, un proveedor de cloud puede ser más crítico para la continuidad que varios departamentos. Una agencia puede haber acumulado conocimiento de marca durante años. Un maintainer de software abierto puede sostener una dependencia que nadie ha inventariado.

La tercera es que reducir headcount vuelve ligera a una organización. A veces solo cambia la cuenta contable. Si una actividad pasa de salarios a servicios adquiridos, los ingresos por empleado aumentan aunque las mismas personas sigan trabajando, ahora bajo otra entidad. El numerador no ha cambiado; el denominador visible sí. Llamarlo productividad es una **reclassification fallacy**.

Las cuentas de producción ofrecen un recordatorio útil. El sistema KLEMS de BEA y BLS explica el crecimiento a través de capital, trabajo, energía, materiales, servicios intermedios y

productividad multifactor, no de payroll aislado.⁵ Una empresa no necesita reproducir una cuenta nacional, pero sí aprender la lección: output tiene más de un input.

Lo que el organigrama deja fuera

Un resultado empresarial puede depender de al menos seis capas:

1. **trabajo empleado**, con una relación laboral directa;
2. **trabajo no empleado**, desde founders hasta contractors, firmas profesionales y personal de proveedores;
3. **capital y software**, incluidos sistemas que amplifican o sustituyen partes del trabajo;
4. **inputs intermedios**, materiales, datos y servicios adquiridos;
5. **infraestructura compartida**, mercados, pagos, cloud, estándares y redes;
6. **instituciones**, permisos, cumplimiento, confianza y mecanismos de responsabilidad.

Las capas no son equivalentes. Un sistema de IA no tiene derechos laborales; un especialista no es una API; un proveedor no forma parte de la cultura por aparecer en un contrato. El objetivo de hacerlas visibles no es mezclarlas, sino evitar que una sola categoría capture todo el concepto de capacidad.

El Bureau of Labor Statistics ofrece otra advertencia metodológica. Su suplemento de julio de 2023 estimó 11,9 millones de independent contractors en el empleo principal o único, el 7,4% del empleo total de aquella medición.⁶ La cifra es valiosa precisamente porque su definición es estrecha: no pretende contar toda persona que haya realizado trabajo adicional, vendido servicios

mediante una entidad o participado en una cadena de proveedores. Cambie la definición y cambiará el número. Una “workforce externa” sin definición puede crecer o encogerse al gusto de quien la presenta.

Capability Baseline

Antes de celebrar que una empresa es pequeña, distribuida o automatizada, necesitamos una línea base. El **Capability Baseline** no intenta medir con exactitud imposible cada minuto de trabajo. Obliga a declarar el denominador suficiente para no engañarnos.

Para un outcome material —por ejemplo, cerrar el mes contable, entregar una funcionalidad, atender una categoría de clientes o fabricar una unidad— registra:

- qué resultado debe producirse y con qué estándar;
- qué empleo directo participa;
- qué otro trabajo humano contribuye, aunque solo pueda estimarse como rango;
- qué capital, software y datos son necesarios;
- qué inputs y proveedores son críticos;
- quién decide, quién revisa y quién responde;
- qué sucede cuando una pieza falla.

El baseline no produce una medalla. Puede mostrar que una arquitectura externa es más rápida y robusta; también puede revelar una dependencia, un coste oculto o un founder trabajando de manera insostenible. Su función es disciplinar la comparación.

Suponga que dos empresas generan el mismo ingreso con veinte

empleados. Una ha construido software propio y opera una red de especialistas bien documentada. La otra depende de un fundador, de jornadas invisibles y de un proveedor único que puede cambiar precio o disponibilidad. En una tabla de revenue per employee parecen gemelas. Como sistemas de capacidad son radicalmente distintas.

La objeción de la falsa precisión

Hay una buena razón para no reconstruir cada cadena de valor en cada decisión: cuesta. La contabilidad existe para simplificar. Un mapa que intenta incluir a todo participante puede volverse tan complejo que deje de orientar. Peor aún, estimaciones débiles de horas externas pueden producir una apariencia científica sin sustancia.

La respuesta no es reemplazar payroll como métrica administrativa. Es escoger el denominador de acuerdo con la afirmación. Para calcular obligaciones laborales directas, payroll es central. Para comparar productividad del sistema, no basta. Para decidir si una capacidad es resiliente, necesitamos proveedores, dependencias y recovery. Para valorar una “empresa de diez personas”, necesitamos trabajo y capital ocultos.

La regla puede expresarse así: **cuanto más extraordinaria sea la afirmación sobre ligereza, escala o autonomía, más completo debe ser el denominador.**

Una decisión distinta

La mayoría de los procesos de planificación comienza con una lista de puestos. Pruebe a comenzar con una lista de outcomes:

- ¿Qué debe ser verdad al final del trimestre?
- ¿Qué capacidad permite producirlo de forma repetible?
- ¿Con qué frecuencia se necesita?
- ¿Qué continuidad, aprendizaje y control exige?
- ¿Qué trabajo y capital lo sostienen hoy?

Todavía no estamos eligiendo entre contratar, acceder o automatizar. Estamos evitando que la elección ocurra de forma inconsciente. El cambio parece pequeño: pasar de “¿a quién contratamos?” a “¿qué debe poder hacer el sistema?”. En realidad altera la unidad básica de diseño.

Payroll seguirá siendo una de las instituciones más importantes de la empresa. Pero deja de ser su perímetro intelectual. El organigrama muestra relaciones formales; el sistema de capacidades muestra lo que la organización puede hacer y de qué depende.

La pregunta que abre el siguiente capítulo es inevitable: **si payroll es solo una frontera, ¿dónde termina realmente el talento de una empresa?**

IMPLICACIÓN EJECUTIVA

Antes de autorizar la próxima contratación material, seleccione el outcome al que responde y complete un Capability Baseline. No utilice el ejercicio para rechazar el puesto; utilícelo para descubrir qué pregunta estaba resolviendo el puesto por defecto.

Tres cuentas para una sola empresa

El problema de la *Payroll Company* no es que cuente mal la nómina. Es que una cifra correcta se utiliza como sustituto de otras dos cuentas que responden preguntas distintas. La primera es la **cuenta laboral**: quién mantiene una relación de empleo con la entidad, con qué obligaciones y bajo qué jurisdicción. La segunda es la **cuenta económica de producción**: cuánto trabajo humano, capital, software, datos, materiales y servicios intermedios se necesitan para generar el resultado. La tercera es la **cuenta de capacidad**: qué outcomes puede producir la organización con fiabilidad, qué dependencias sostienen esa facultad y cuánto tardaría en recuperarla si una pieza desapareciera.

Las tres cuentas se solapan, pero no son intercambiables. La cuenta laboral permite gobernar derechos, remuneración, desarrollo y obligaciones del empleador. La cuenta económica impide llamar productividad a una mera reclasificación del trabajo. La cuenta de capacidad muestra si el sistema puede cumplir, aprender y recuperarse. Una empresa puede mejorar una de ellas y empeorar otra. Reducir payroll puede bajar costes fijos y, al mismo tiempo, aumentar gasto externo, fragilidad o carga del fundador. Contratar un equipo puede elevar el coste fijo y reducir la latencia, el retrabajo y el riesgo operativo. Automatizar un paso puede disminuir horas de ejecución y aumentar la necesidad de assurance, datos y supervisión.

Esta separación corrige una confusión frecuente en consejos y comités de inversión. Cuando alguien afirma que una compañía “escala sin personas”, hay que preguntar a qué cuenta se refiere. Puede estar diciendo que crece sin aumentar empleo directo. Eso es observable. No demuestra que crezca sin más trabajo humano en proveedores, clientes, comunidades o fundadores, ni sin más capital y cómputo. Tampoco demuestra que su capacidad aumente: una

compañía puede atender más volumen en condiciones normales y volverse menos recuperable ante un fallo crítico.

La misma disciplina sirve en sentido contrario. Una organización con mucho headcount no posee necesariamente más capacidad. Puede mantener puestos duplicados, sistemas que impiden ejecutar o conocimiento encerrado en unidades que no colaboran. El tamaño laboral expresa un compromiso institucional importante; no garantiza que ese compromiso esté convertido en outcomes. Las tres cuentas obligan a abandonar dos slogans simétricos: “más personas significa más capacidad” y “menos personas significa más productividad”.

El empleo como opción institucional

Una relación laboral estable suele evaluarse como coste recurrente. Esa descripción es incompleta. También compra una **opción de coordinación**. La empresa puede reasignar atención dentro de límites razonables, invertir en conocimiento cuyo uso exacto todavía no conoce y convocar una respuesta cuando aparece una excepción. No necesita negociar un contrato nuevo para cada incidencia. A cambio asume obligaciones, sostiene capacidad durante periodos de menor demanda y acepta que la relación no es un recurso que pueda encenderse y apagarse sin consecuencias.

Esta opción explica por qué el empleo sigue siendo eficiente incluso cuando una tarea aislada podría contratarse externamente a menor precio. El objeto de comparación no es la tarea prevista, sino la distribución completa de situaciones que la organización deberá resolver. En un workflow estable y perfectamente especificable, la modalidad externa puede funcionar con una interfaz ligera. En un dominio que cambia, donde los casos raros importan y el conocimiento se acumula, la relación estable puede absorber

incertidumbre mejor que una sucesión de encargos.

El error habitual consiste en valorar la capacidad interna por su utilización media y la externa por su precio marginal. Si un especialista interno dedica solo una parte de la semana a la cuestión crítica, parece infrautilizado. Pero quizá su valor reside en estar disponible cuando ocurre el cinco por ciento de casos que determina seguridad, reputación o continuidad. Del mismo modo, una agencia contratada por proyecto puede parecer variable aunque exija un volumen mínimo, una transferencia de contexto reiterada y dirección senior no registrada.

Por eso el **Capability Baseline** debe incluir valor de opción y coste de indisponibilidad. Dos preguntas lo hacen visible: ¿qué decisión no podríamos tomar si esta relación desapareciera mañana?, y ¿qué coste asumimos por mantenerla preparada aunque hoy no esté al máximo de utilización? La primera evita vaciar el sistema. La segunda evita proteger toda estructura existente bajo la etiqueta de resiliencia.

Del presupuesto de puestos al presupuesto de outcomes

Cambiar la unidad de planificación no significa eliminar el presupuesto de plantilla. Significa colocar antes una capa de decisiones. Para cada prioridad material, el equipo directivo puede construir una ficha con cinco campos: outcome, estándar de aceptación, demanda esperada, consecuencias del fallo y capacidades requeridas. Solo después asigna modalidades, derechos, gasto y responsables.

La secuencia altera la conversación. “Contrataremos a tres analistas” se convierte en “necesitamos producir previsiones con

esta frecuencia, precisión y capacidad de explicar desviaciones”. Esa formulación puede seguir conduciendo a tres contrataciones. También puede revelar que el cuello de botella está en los datos, que una parte requiere criterio interno, que otra admite un proveedor recurrente o que varias tareas pueden automatizarse bajo revisión. El framework no está diseñado para reducir puestos; está diseñado para impedir que el puesto sea una respuesta anterior a la pregunta.

Un presupuesto de outcomes contiene, como mínimo, cuatro reconciliaciones. La primera conecta cada outcome con responsables y trabajo humano total. La segunda conecta ese trabajo con software, capital e inputs. La tercera registra el coste de integración, assurance y recuperación, no solo las partidas de compra. La cuarta muestra qué conocimiento debe quedar después de ejecutar. Si la empresa paga por resolver el mismo problema cada trimestre, el gasto externo no ha construido capacidad; ha alquilado una respuesta repetida.

En una empresa pequeña este ejercicio puede caber en una hoja. En una grande puede incorporarse al ciclo de planificación y a las revisiones de portfolio. No requiere estimar cada minuto ni crear una taxonomía perfecta. Requiere suficiente visibilidad para que una afirmación sobre eficiencia, riesgo o escala utilice el denominador correcto.

Lo que debería ver el consejo

El consejo no necesita un inventario de todos los freelancers, contratos y aplicaciones. Sí necesita saber si los outcomes críticos descansan sobre dependencias que la contabilidad de headcount oculta. Cuatro vistas suelen ser suficientes:

1. capacidades cuya pérdida detendría una prioridad estratégica;
2. concentración de trabajo, proveedor, plataforma, dato o persona;
3. tiempo de sustitución y recuperación bajo un escenario plausible;
4. conocimiento y derechos que permanecen en la empresa al finalizar una relación.

Estas vistas conectan workforce con riesgo empresarial. Una dependencia cloud, una librería mantenida por una comunidad, una firma que concentra assurance o una persona que traduce todo el contexto pueden ser más materiales que una variación moderada de plantilla. Hacerlas visibles no implica internalizarlas. Permite elegir si conviene aceptar la dependencia, reservar una alternativa, documentar una interfaz, reforzar un contrato o reconstruir capacidad propia.

La *Payroll Company* empieza a cambiar cuando puede explicar su capacidad sin utilizar headcount como atajo. La nómina conserva su función legal, humana y organizativa. Lo que pierde es el monopolio sobre la descripción de la empresa.

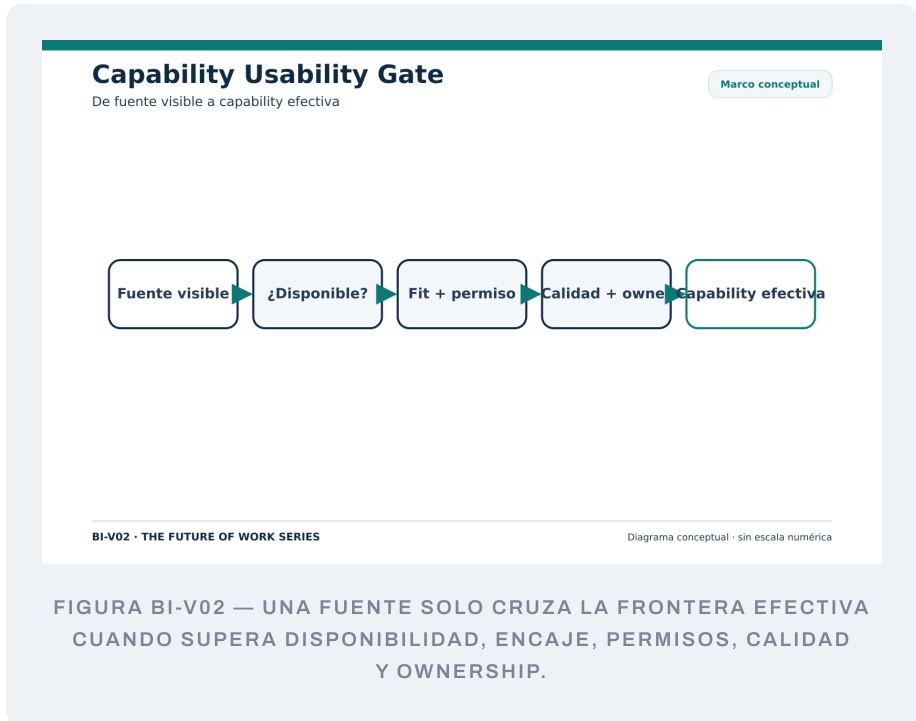
Payroll cuenta relaciones laborales. No cuenta, por sí solo, toda la capacidad ni todo el trabajo de una empresa.

NOTAS DEL CAPÍTULO

1. Apple, **Supply Chain** y **SEC Filings**. La referencia acredita una red de producción y obligaciones directas; no permite calcular trabajo total sin datos adicionales.
2. U.S. Census Bureau, **Nonemployer Statistics**, datos más recientes disponibles: 2023.
3. Ronald H. Coase, “**The Nature of the Firm**”, *Economica*, 1937.
4. Alfred D. Chandler Jr., *The Visible Hand*, Harvard University Press, 1977.
5. BEA-BLS, **Integrated Industry-Level Production Account (KLEMS)**.
6. U.S. Bureau of Labor Statistics, *Contingent and Alternative Employment Arrangements — July 2023*, publicado en 2024. La medida se refiere al empleo principal o único.

Capítulo 2

THE TALENT BOUNDARY



De la frontera mental a la frontera efectiva

La NASA conoce una clase particular de humildad. Sus problemas pueden ser extraordinariamente difíciles y, al mismo tiempo, la persona capaz de desbloquear uno de ellos puede no trabajar en la agencia ni en ninguna de sus empresas habituales. El NASA Tournament Lab existe para conectar problemas públicos con redes externas de problem-solvers mediante retos, premios y plataformas.¹

La historia suele contarse así: una institución abre un problema y el talento del mundo responde. Es una buena historia, pero termina demasiado pronto. Una solución ganadora todavía debe entenderse, verificarse, adaptarse, integrarse y mantenerse. Alguien debe decidir si puede operar en un sistema real. Alguien debe responder si falla. La convocatoria demuestra que una idea puede llegar desde fuera; no demuestra que toda capacidad esté disponible bajo demanda.

Entre publicar un reto y depender de su resultado hay una frontera.

Dos errores simétricos

El primer error es creer que el talento de una organización coincide con su nómina. La división administrativa entre HR y procurement refuerza esa intuición: empleados a un lado, proveedores al otro. Sin embargo, ambos pueden contribuir al mismo workflow, usar los mismos datos y afectar al mismo cliente.

El segundo error aparece cuando intentamos corregir el primero. Si internet permite ver millones de perfiles, miles de firmas y una oferta creciente de software, resulta tentador declarar que toda esa capacidad está “a nuestro alcance”. Visibilidad se convierte en acceso; acceso, en disponibilidad; disponibilidad, en productividad. Cada salto elimina una fricción que sigue ahí.

Una empresa puede encontrar cien especialistas y no conseguir a ninguno durante la semana crítica. Puede contratar a una firma excelente y tardar meses en transferir contexto. Puede conectar un modelo de IA a un proceso y descubrir que no tiene permisos, datos fiables ni un método para revisar errores. El catálogo es real. La capacidad operativa todavía no.

Necesitamos distinguir tres cosas.

La **frontera contractual** reúne relaciones y activos sobre los que la empresa tiene derechos formales: empleo, propiedad, contratos y decision rights. El **conjunto accesible** contiene fuentes que, en principio, podrían utilizarse: redes, marketplaces, proveedores, comunidades, software o sistemas de IA. La **frontera efectiva de capacidad** comprende aquello que puede producir un outcome específico por encima de umbrales aceptables de disponibilidad, integración, calidad, aprendizaje y recuperación.

Estas fronteras no forman círculos perfectos. Una comunidad puede aportar soluciones sin pertenecer a la frontera contractual. Un proveedor contratado puede estar formalmente dentro de la red de contratos y, sin embargo, no estar disponible durante un shock. Un empleado puede estar en payroll pero carecer de la capacidad necesaria para un problema nuevo. Las fronteras se solapan y cambian con el tiempo.

El Capability Usability Gate

Para que una fuente pase del conjunto accesible a la frontera efectiva debe atravesar seis gates. Los llamaremos **AIALRR** por sus términos en inglés, aunque lo importante no es el acrónimo sino la secuencia.

AVAILABILITY

¿La capacidad puede comprometerse cuando se necesita, con la latencia y duración requeridas? “Existe en el mercado” no responde si aceptará el encargo, si tiene conflicto o si coincide con nuestro calendario.

INTEGRATION

¿Puede incorporarse al workflow con el contexto, datos, herramientas, permisos e interlocutores disponibles? Una solución valiosa puede ser difícil de convertir en operación.

ASSURANCE

¿Podemos comprobar calidad, seguridad, cumplimiento y fiabilidad? ¿Qué significa “bueno” antes de recibir el output? ¿Qué errores severos importan?

LEARNING

¿La organización retendrá conocimiento o pagará de nuevo por descubrir lo mismo? ¿Quién documenta decisiones, supuestos y excepciones? El aprendizaje no aparece automáticamente al contratar expertise.

RESPONSIBILITY

¿Quién posee el decision right, acepta el output y responde por sus consecuencias? Externalizar ejecución no elimina la responsabilidad institucional.

RECOVERY

¿Qué ocurre si la capacidad desaparece, se equivoca o debe sustituirse? ¿Existe fallback, redundancia, exportabilidad y tiempo de recuperación aceptable?

El gate convierte una metáfora espacial —la frontera— en preguntas observables. Una firma puede decidir que una fuente es efectiva para diseño exploratorio y no para operación crítica. La misma fuente puede estar dentro en una geografía y fuera en otra. La clasificación siempre lleva capacidad, fecha y contexto.

P&G y la apertura con estructura

Procter & Gamble describe Connect + Develop como un proceso mediante el cual equipos de la compañía buscan innovadores, patentes y tecnologías externas y construyen partnerships para desarrollar soluciones.² El caso importa porque no equivale apertura a ausencia de capacidad interna. Buscar fuera requiere formular necesidades, evaluar propuestas, gestionar propiedad intelectual, adaptar una solución y conectarla con manufactura, marca o distribución.

Open innovation no convierte la empresa en un recipiente vacío. Cambia dónde puede originarse el conocimiento y eleva el valor de las capacidades de absorción e integración. Cuanto mayor sea el conjunto accesible, mayor puede ser el trabajo de selección.

P&G y NASA muestran dos rutas distintas. Un reto puede producir una solución discreta para un problema especificable. Una red de innovación repetida necesita relaciones, criterios y mecanismos de transferencia. En ambos casos el valor no se encuentra únicamente “fuera”. Aparece en la interfaz entre quienes plantean, quienes resuelven y quienes integran.

Kaggle y la diferencia entre solución y capacidad

Una competición de datos puede encontrar un modelo de gran rendimiento. Pero un notebook ganador no es todavía un sistema de producción. Puede faltar documentación, robustez ante datos nuevos, seguridad, monitoring, coste operativo o una persona responsable de mantenerlo. El problema no invalida la competición; aclara qué compramos.

Quizá accedimos a una **solución**. Para obtener una **capacidad** necesitamos repetir el outcome, adaptarlo, detectar drift y

recuperarnos de fallos. Confundir ambos objetos produce proyectos que deslumbran en la demo y se degradan en operación.

Esta distinción sirve también fuera de la tecnología. Un asesor puede entregar un diagnóstico excelente sin convertirse en capacidad continua de decisión. Una agencia puede ejecutar una campaña sin transferir conocimiento. Una comunidad puede descubrir una idea que la empresa no sabe adoptar. La frontera efectiva no premia la procedencia; exige el sistema completo.

Falsos positivos y falsos negativos

Los mapas de talento suelen producir falsos positivos: fuentes dibujadas dentro que no superan los gates. La plataforma tiene perfiles, pero no disponibilidad. El proveedor tiene certificaciones, pero no conoce el contexto. El modelo responde, pero no puede usar las herramientas necesarias. La persona es excelente, pero el contrato impide el acceso a datos.

También producen falsos negativos. Una relación externa repetida puede haber acumulado más contexto que parte de la plantilla. Un maintainer de open source puede influir en la continuidad de un producto sin aparecer en procurement. Un cliente experto puede aportar conocimiento que no existe dentro. Ignorar estas relaciones no las hace menos críticas; solo las vuelve menos gobernadas.

El Capability Boundary Map utiliza cinco estados:

- **controlled:** la empresa conserva control estable suficiente;
- **committed:** existe reserva, continuidad y memoria relacional;
- **effective, not controlled:** puede producir ahora, aunque el control sea limitado;
- **accessible only:** visible o contratable en principio, todavía sin superar gates;
- **out of reach:** no disponible por coste, geografía, permiso, escasez o madurez técnica.

El mapa no es una taxonomía eterna. Debe llevar fecha. Una regulación puede sacar una fuente de alcance. Un acuerdo marco puede moverla de accesible a committed. Un incidente puede demostrar que una capacidad que creíamos efectiva carecía de recovery.

¿Una tautología elegante?

Hay una objeción seria. Si cada empresa elige sus propios umbrales, “capacidad efectiva” podría significar simplemente “lo que funciona”, una etiqueta retrospectiva sin poder de decisión.

El remedio es comprometer los umbrales antes de clasificar. Para atención al cliente, por ejemplo: disponibilidad en determinada franja; tiempo máximo de ramp; resolución y calidad mínimas; cero acceso no autorizado; owner interno; plan de salida. Una fuente entra o queda fuera con evidencia. Si falla de forma sistemática en ramp, calidad o recovery, el mapa se corrige.

El objetivo no es crear precisión universal, sino hacer explícita una inferencia que normalmente queda oculta. “Podemos conseguirlo fuera” se transforma en seis preguntas. A menudo la conversación más valiosa aparece cuando una de ellas no tiene

respuesta.

La frontera es una capacidad de gobierno

Ampliar la frontera efectiva puede crear ventaja: más variedad, rapidez o especialización. También puede crear una superficie mayor de dependencia y responsabilidad. Cada nueva fuente añade relaciones, permisos, criterios y caminos de fallo. La organización que accede a más no necesariamente puede hacer más. La organización que integra y recupera mejor, sí puede.

Por eso no conviene medir la frontera por el número de perfiles, partners o herramientas. Médala por outcomes que pueden producirse con fiabilidad y por el coste total de hacerlos efectivos. Una red pequeña, comprometida y bien conectada puede ofrecer más capacidad que un marketplace gigantesco sin contexto.

El capítulo comenzó preguntando dónde termina el talento de una empresa. La respuesta provisional es incómoda: no termina en el contrato, pero tampoco llega hasta todo lo visible. Termina, para cada outcome, donde una fuente supera los gates y la institución puede aprender y responder.

Ahora aparece una dificultad ética y analítica. Si una parte de esa capacidad reside en personas que no están empleadas, no podemos decir que la empresa las “posee”. **¿Qué controla realmente una firma y qué diferencia una relación comprometida de un acceso meramente nominal?**

IMPLICACIÓN EJECUTIVA

Elija una capacidad crítica y coloque cada fuente actual en los cinco estados. Escriba al lado qué evidencia la hace efectiva y qué evento la expulsaría. Si no puede formular el criterio, todavía tiene un inventario, no una frontera.

Una frontera distinta para cada outcome

Hablar de “la” frontera de talento sugiere un perímetro único. En la práctica, la organización tiene tantas fronteras efectivas como outcomes materiales. Una firma puede movilizar diseño gráfico en cuarenta y ocho horas y ser incapaz de incorporar con seguridad a un especialista de datos en el mismo plazo. Puede acceder a asesoramiento jurídico en varios países y no disponer de una relación preparada para responder a un incidente durante el fin de semana. Puede contratar un modelo avanzado por API y carecer de los datos, permisos o pruebas que convertirían esa conexión en una capacidad operacional.

La frontera, por tanto, debe llevar tres apellidos: **capacidad, contexto y fecha**. Sin ellos, el mapa confunde potencial con operación. “Tenemos acceso a ciberseguridad” no es una afirmación útil. “Para investigar un incidente de severidad alta en nuestra infraestructura europea, tenemos un partner comprometido, acceso autorizado, un tiempo de respuesta acordado y un owner interno” sí lo es. La primera frase describe una categoría de gasto. La segunda describe una capacidad efectiva y falsable.

Esta precisión cambia el uso ejecutivo del concepto. La pregunta deja de ser cuántos proveedores o perfiles tiene la empresa y pasa a

ser qué outcomes puede prometer. Un marketplace amplía el conjunto accesible; un acuerdo marco puede reducir el tiempo de contratación; una relación repetida aporta contexto; una interfaz probada reduce el *ramp time*; un plan de salida mejora recuperación. Cada intervención mueve una dimensión diferente. Ninguna, por sí sola, completa la frontera.

Acceso, reserva y opción

Una fuente visible ofrece una opción nominal. Para que esa opción tenga valor debe poder ejercerse. La diferencia se parece a la que existe entre conocer un hotel y tener una habitación reservada en la noche de mayor demanda. En condiciones normales ambas parecen equivalentes; durante el pico, no.

Las organizaciones sobrevaloran a menudo la oferta spot porque observan el mercado cuando no lo necesitan. Ven perfiles, demos y proveedores disponibles y extrapolan esa abundancia al momento crítico. Pero la demanda de muchas capacidades está correlacionada. Tras una brecha de seguridad, una nueva regulación o un cambio tecnológico, numerosas empresas buscan a los mismos especialistas. La profundidad nominal desaparece justo cuando más importa. La frontera efectiva debe probarse bajo el estado del mundo en el que se utilizará, no solo bajo el estado en que se dibuja.

El compromiso compra prioridad, pero tampoco garantiza todo. Un retainer puede reservar horas sin reservar a la persona concreta. Un proveedor puede prometer respuesta y depender a su vez de un tercero. Un modelo puede estar disponible técnicamente y cambiar de versión, coste o política. Por eso el gate de disponibilidad debe preguntar no solo “¿puede empezar?”, sino “¿qué recurso exacto está comprometido, bajo qué condiciones y contra qué demanda competirá?”.

La opción también caduca. Un especialista que conocía la arquitectura hace un año puede necesitar un onboarding nuevo. Un acuerdo puede seguir vigente y sus supuestos regulatorios no. Una integración puede romperse por cambios de API. El mapa necesita fecha de revisión y señales de degradación. Sin mantenimiento, la frontera efectiva se convierte lentamente en ficción administrativa.

Cuatro pruebas de choque

Una frontera se comprende mejor cuando se intenta romperla. Antes de declarar una capacidad efectiva, sométala a cuatro shocks.

El primero es un **shock de volumen**. ¿Qué ocurre si la demanda se duplica durante dos semanas? Una fuente que funciona a cadencia normal puede no ampliar capacidad. El segundo es un **shock de ausencia**. ¿Puede operar el sistema si la persona o proveedor preferido no está? El tercero es un **shock de confianza**: un error severo, una brecha o un conflicto obliga a retirar permisos y revisar outputs. El cuarto es un **shock institucional**: cambia una norma, una jurisdicción o la interpretación de la relación, y la modalidad prevista deja de ser viable.

Los shocks no necesitan convertirse en simulacros costosos para toda actividad. Una revisión de sobremesa puede descubrir dependencias importantes. ¿Dónde está la documentación? ¿Quién tiene credenciales? ¿Existe un criterio de aceptación independiente del proveedor? ¿Podemos exportar datos y artefactos? ¿Quién decide detener el workflow? ¿Cuánto tarda la sustitución? Si las respuestas dependen de una sola persona, la fuente quizá produce hoy, pero la capacidad pertenece menos a la organización de lo que el mapa indica.

El resultado no siempre será internalizar. Puede bastar con dividir

una dependencia, construir un fallback, reservar disponibilidad, limitar permisos o conservar una competencia de assurance dentro. Lo importante es que la frontera se define también por la recuperación, no solo por el estado estable.

La frontera del conocimiento

Disponibilidad y calidad visible son insuficientes cuando el outcome depende de aprender. Una consultora puede entregar una recomendación excelente y dejar a la organización incapaz de reconstruir el razonamiento. Un proveedor de software puede operar un proceso y conservar el historial de excepciones en sistemas que el cliente no puede exportar. Un experto puede resolver incidencias repetidas sin que cambien la documentación ni el diseño.

En esos casos la ejecución cruza la frontera, pero el aprendizaje no vuelve. La empresa amplía output a corto plazo y estrecha capacidad futura. La decisión correcta no exige que todo conocimiento quede codificado —parte es tácito y parte pertenece legítimamente a la contraparte—, pero sí que la organización identifique qué memoria necesita conservar para formular, evaluar y recuperar.

El gate de aprendizaje puede comprobarse con artefactos concretos: decisiones y supuestos registrados, cambios incorporados a procesos, datos de aceptación, excepciones clasificadas, responsables capaces de explicar el resultado y una salida que transfiera lo acordado. “Hemos trabajado muchas veces juntos” no basta si toda la memoria reside al otro lado.

La frontera de responsabilidad

La organización puede distribuir ejecución, pero no debería dibujar responsabilidad como si fuera otra mercancía intercambiable. Un proveedor puede asumir obligaciones contractuales; un profesional puede responder por su trabajo; un sistema puede registrar decisiones. Aun así, la entidad que promete el producto, trata los datos, atiende al cliente o ocupa una posición regulada conserva deberes que no desaparecen al mover tareas.

Por eso la frontera efectiva contiene una pregunta institucional: ¿quién tiene autoridad para aceptar, rechazar, escalar y detener? Si el owner interno carece de criterio para evaluar el output, la responsabilidad es nominal. Si tiene criterio pero no acceso a evidencia, es impotente. Si la contraparte controla el proceso de facto y la empresa solo firma, el diseño puede haber externalizado el juicio que necesitaba conservar.

La responsabilidad convierte el mapa de talento en un mapa de gobierno. Una fuente entra en la frontera efectiva cuando puede producir y cuando la institución puede decidir qué hacer con lo producido. Esta condición es especialmente importante para IA: una respuesta rápida no constituye capacidad si nadie ha definido los errores intolerables, los permisos y el fallback.

Gobernar el movimiento, no congelar el perímetro

El propósito del *Capability Boundary Map* no es fijar una frontera perfecta. Es registrar por qué una fuente está dentro y qué haría moverla. Una capacidad accesible puede volverse committed después de varias colaboraciones y una reserva explícita. Una capacidad controlled puede perder efectividad porque el conocimiento se ha quedado obsoleto. Un proveedor recurrente puede salir tras un incidente. Una herramienta puede entrar para tareas de bajo riesgo y permanecer fuera de decisiones críticas.

Cada entrada debería tener un **evento de promoción**, un **evento de degradación** y un **evento de salida**. Promoción: qué evidencia permitiría confiar más. Degradación: qué señal obliga a reducir alcance o reforzar controles. Salida: qué condición vuelve inaceptable la fuente. Esos tres campos transforman el mapa en una herramienta de gestión dinámica.

La ventaja no procede de dibujar una frontera más grande. Procede de moverla con mayor criterio: ampliar donde la organización puede integrar, estrechar donde el riesgo supera el valor y convertir experiencia en mejores gates. Una empresa abierta sin esta disciplina acumula opciones nominales. Una empresa deliberada construye capacidad efectiva.

Internet amplía lo visible. Solo la disponibilidad, la integración, la verificación, el aprendizaje, la responsabilidad y la recuperación amplían la capacidad efectiva.

NOTAS DEL CAPÍTULO

1. NASA, [NASA Tournament Lab](#) y [Center of Excellence for Collaborative Innovation](#).
2. Procter & Gamble, [Innovation — Connect + Develop](#). Es una descripción corporativa del mecanismo; no prueba por sí sola su impacto causal.

The best talent doesn't need to work for you to work with you.

CONTROLLED, COMMITTED, ACCESSIBLE



40

Puede poseer activos. Puede conservar derechos sobre una tecnología. Puede definir decision rights, reservar disponibilidad y proteger información. Puede construir una relación laboral que compromete tiempo y establece autoridad dentro de límites jurídicos. Pero hablar de talento “poseído” mezcla personas, habilidades y derechos de control en una sola categoría.

Corregir el lenguaje no es solo una cuestión ética. Mejora la decisión. Dos capacidades externas pueden parecer idénticas en procurement y comportarse de forma opuesta: una firma asociada que trabaja con nosotros cada semana y un especialista encontrado ayer en un marketplace. Dos empleados pueden compartir contrato y ofrecer grados distintos de disponibilidad o conocimiento. “Interno/externo” describe una relación formal; no agota el sistema operativo.

La taxonomía más útil distingue **controlled**, **committed** y **accessible capability**.

Controlled: lo que la empresa puede gobernar de forma estable

Una capacidad está bajo control estable cuando la organización conserva suficientes decision rights, disponibilidad, activos, procesos y mecanismos de aprendizaje para orientar su uso y responder por ella. El empleo puede ser una de sus bases, pero no la única. La propiedad intelectual, una licencia, infraestructura propia, un contrato de largo plazo o una arquitectura que permite sustituir componentes también pueden aportar control.

La teoría de property rights recuerda por qué importa. En contratos incompletos no es posible prever cada contingencia. Quien conserva derechos residuales de control puede decidir qué hacer

cuando aparece una situación no especificada.¹ Esa capacidad influye en la inversión: una parte invertirá menos si teme que otra capture el valor una vez realizada la inversión.

Control ofrece continuidad y apropiación. También cuesta. Crea estructura fija, inversión, obligaciones y, a veces, rigidez. Una empresa puede internalizar una capacidad para evitar dependencia y terminar aislada de mejor expertise. El control es una elección de governance, no una insignia de seriedad.

Committed: el espacio olvidado

Entre control estable y mercado abierto existe un territorio amplio. Una persona reserva dos días al mes para una empresa durante años. Una firma especializada mantiene un equipo identificado, conoce la arquitectura y participa en decisiones recurrentes. Un pool de profesionales tiene acuerdos de disponibilidad, estándares y memoria. No están en payroll, pero tampoco son una fuente spot.

Llamaremos **committed capability** a una capacidad externa con compromiso verificable de tiempo, prioridad o continuidad y con acumulación de contexto. Sus propiedades no proceden del título del contrato, sino de la conducta:

- existe una reserva o cadencia;
- la relación se repite;
- se acumula conocimiento específico;
- hay expectativas de respuesta y mecanismos de escalado;
- la salida requiere transferencia;
- ambas partes asumen alguna inversión relacional.

Unilever presentó U-Work como una modalidad en la que

participantes podían recibir un retainer y beneficios mientras realizaban asignaciones cuando estaban disponibles.² El caso es interesante no porque pruebe una forma superior —la información pública no basta para eso—, sino porque rompe la oposición binaria. Compromiso y empleo estándar no son sinónimos; flexibilidad y mercado spot tampoco.

La capa committed puede reducir latencia y conservar memoria. También puede esconder un problema. Si la empresa controla de hecho cómo, cuándo y con qué dependencia trabaja una persona, la etiqueta “independiente” puede no describir la sustancia de la relación. En Estados Unidos, el IRS considera control conductual, control financiero y tipo de relación, y advierte que el nombre del contrato no decide el estatus.³ En la Unión Europea, la regulación de trabajo en plataformas refuerza la atención sobre dirección, control, transparencia y situación laboral.⁴

Una arquitectura responsable no diseña committed capacity para conseguir autoridad laboral sin obligaciones laborales. La diseña para reservar una unidad genuinamente independiente de expertise o ejecución, con scope y autonomía coherentes.

Accessible: opción sin garantía

La capa accessible contiene fuentes identificables que pueden contratarse o activarse en principio, pero sin compromiso suficiente de disponibilidad, contexto o continuidad. Su gran ventaja es la variedad. Permite explorar habilidades raras, geografía nueva o picos de demanda sin sostener permanentemente todas las opciones.

Su gran peligro es la nominalidad. Un marketplace con miles de perfiles no promete que el especialista adecuado acepte el trabajo,

entienda el problema o permanezca disponible. La economía de plataformas reduce costes de búsqueda y matching, pero no elimina reputación, selección, comunicación, protección o dependencia. La investigación sobre mercados de trabajo online encuentra que las relaciones persistentes y la reputación siguen siendo relevantes; el mercado no se convierte en una sucesión sin fricción de transacciones anónimas.⁵

La capacidad accesible es una **opción**. Su valor depende de poder ejercerla. Para una tarea tolerante a latencia y fácilmente verificable puede ser suficiente. Para una decisión urgente y altamente contextual quizá no lo sea.

El mapa y sus overlays

Las tres capas no son categorías de personas. Clasifican una capacidad concreta. Una misma empresa puede mantener bajo control su arquitectura de datos, comprometer expertise de seguridad y acceder de manera spot a traducción. Una misma persona podría ofrecer capacidades distintas bajo relaciones distintas, aunque esa complejidad exige especial cuidado jurídico y operativo.

El mapa necesita al menos seis overlays:

DIMENSIÓN	PREGUNTA
Control residual	¿Quién decide ante una contingencia no prevista?
Reserva	¿Qué disponibilidad está realmente comprometida?
Time-to-ramp	¿Cuánto tarda en producir por encima del umbral?
Contexto	¿Qué conocimiento específico conserva y transfiere?
Switching cost	¿Qué cuesta sustituir o salir?
Riesgo bilateral	¿Quién absorbe demanda variable, inversión y fallo?

No hay una flecha de progreso desde controlled hasta accessible. Mover una capacidad hacia el mercado puede aumentar variedad y reversibilidad mientras reduce aprendizaje o recovery. Moverla hacia control puede proteger conocimiento mientras eleva coste fijo. El punto correcto depende de la capacidad.

Total Access Cost

La comparación suele estar sesgada. El coste de contratar se expresa como salario más cargas; el coste externo, como factura. Uno incluye overhead visible y el otro parece variable y limpio. Para decidir necesitamos **Total Access Cost**, un cálculo simétrico.

Para cualquier modalidad incluya:

- precio o compensación;
- búsqueda, selección y negociación;
- onboarding y transferencia de contexto;
- coordinación y revisión;
- herramientas, seguridad y permisos;
- retrabajo y errores;
- retención o pérdida de conocimiento;
- switching, salida y recovery;
- riesgo transferido a la empresa, a la contraparte o a terceros.

El cálculo no exige monetizarlo todo con falsa precisión. Puede combinar cantidades, rangos y riesgos explícitos. Lo importante es evitar dos ficciones: que el salario sea el coste completo del empleo y que la factura sea el coste completo del acceso.

Una capacidad externa barata por hora puede resultar cara si consume dirección senior y genera retrabajo. Un empleado puede parecer caro frente a una necesidad intermitente y, sin embargo, reducir incidentes y acelerar decisiones porque conserva contexto. La comparación pertenece al sistema, no a la tarifa.

Perspectiva bilateral

Hasta ahora hemos mirado la arquitectura desde la empresa. Esa mirada es insuficiente. Una modalidad “elástica” para el comprador puede significar ingresos impredecibles para quien trabaja. Una relación committed puede ofrecer continuidad o crear dependencia sin protección. Un mercado accesible puede ampliar oportunidades y, al mismo tiempo, desplazar costes de búsqueda, equipo y tiempo no facturado hacia los profesionales.

La OIT ha documentado cómo las plataformas pueden abrir fuentes de trabajo e ingreso mientras crean déficits de trabajo decente, gestión algorítmica y transferencia de costes.⁶ No todas las relaciones externas son plataforma ni toda independencia es precariedad. La regla más rigurosa es otra: **no describa un beneficio de la firma sin identificar quién paga por él.**

Por eso cada mapa de capacidad debe incluir riesgo bilateral:

- ¿quién absorbe una caída de demanda?
- ¿quién paga el tiempo de preparación?
- ¿quién financia herramientas y formación?
- ¿quién puede salir y con cuánto aviso?
- ¿qué concentración de ingreso o proveedor existe?
- ¿qué voz tiene cada parte sobre reglas y evaluación?

La arquitectura que ignora estas preguntas puede ser eficiente durante un trimestre y frágil durante años.

Cuando las categorías se vuelven excusas

Una empresa podría llamar committed a cualquier proveedor favorito, controlled a cualquier función que quiera proteger y accessible a todo lo demás. El framework se volvería una legitimación elegante de la situación actual.

Para evitarlo, clasifique por evidencia. ¿Existe un derecho de decisión? ¿Una reserva? ¿Una obligación de transferencia? ¿Un historial de respuesta? ¿Puede la contraparte rechazar trabajo? ¿Quién controla el método? ¿Qué ocurre al salir? La conducta desplaza al label.

También hay que aceptar híbridos. Una capacidad puede tener arquitectura controlled, ejecución committed y picos accessible. Seguridad puede conservar políticas y decision rights dentro, usar un partner recurrente para monitorización y contratar una prueba especializada. La unidad sigue siendo la capacidad, no el proveedor.

La pregunta que importa

No necesitamos decidir si lo externo es mejor. Necesitamos saber qué relación estamos construyendo y qué propiedades compra. Control estable compra más que disponibilidad. Compromiso compra más que una hora reservada. Acceso compra una opción, no un resultado.

Una vez que abandonamos la falsa propiedad de las personas, aparece una arquitectura más precisa. La firma puede controlar activos, decisiones y aprendizaje; comprometer capacidad externa; acceder a opciones adicionales. Cada capa tiene un coste, una distribución de riesgo y una forma de fallo.

El mapa todavía no nos dice cuál elegir. Ese es el siguiente problema. **¿Cuándo conviene desarrollar, contratar, acceder o automatizar una capacidad, y cómo evitar que la respuesta se convierta en ideología?**

IMPLICACIÓN EJECUTIVA

Tome cinco capacidades materiales. Para cada una declare la capa actual, la evidencia que la justifica, su Total Access Cost, el riesgo bilateral y el trigger que obligaría a cambiar de modalidad.

El compromiso es un activo, no una zona gris

La capa *committed* puede parecer un punto intermedio cómodo entre empleo y mercado. Su importancia es mayor. Representa un activo relacional que ninguna de las dos categorías extremas describe bien. La empresa obtiene una disponibilidad más previsible, conocimiento específico y un historial de cooperación. La contraparte obtiene continuidad, mejor información sobre demanda y la posibilidad de invertir en una relación cuyo retorno no depende de una sola transacción.

Ese activo se construye. Repetir encargos no basta si cada uno comienza desde cero. El compromiso aparece cuando ambas partes realizan inversiones que mejoran el siguiente outcome: documentación compartida, estándares, acceso preparado, conocimiento de excepciones, una cadencia de decisión y mecanismos de escalado. Una agencia que conoce la marca durante tres años no es simplemente “un proveedor”; tampoco se convierte por ello en una unidad interna. Es una relación con memoria, dependencia y derechos que deben hacerse visibles.

La economía de esa relación incluye una prima de disponibilidad. Si la empresa espera prioridad durante un pico, alguien debe reservar capacidad o renunciar a venderla a otro cliente. Ignorar esa opción produce contratos que prometen respuesta sin financiarla. La tarifa aparente es baja porque la contingencia se desplaza hacia la contraparte. Cuando llega la demanda, la empresa descubre que compró horas posibles, no capacidad comprometida.

El diseño responsable especifica qué se reserva: personas identificadas, un número de horas, una franja de respuesta, un nivel de servicio o una combinación. También declara lo que no se reserva. La claridad protege a ambas partes. La empresa puede

construir fallback para lo que queda fuera; la contraparte puede gestionar su portfolio sin una obligación implícita e ilimitada.

Dependencia en ambos sentidos

Los mapas convencionales registran dependencia del proveedor: concentración, switching cost o propiedad de datos. Rara vez registran la dependencia inversa. Un profesional puede obtener la mayor parte de sus ingresos de una empresa. Una firma pequeña puede invertir en herramientas o formación específicas. Un equipo externo puede dedicar tanto contexto a un cliente que su salida sea costosa aunque el contrato permita terminar con rapidez.

La dependencia bilateral cambia la gobernanza. Si la empresa tiene mucha autoridad operativa y la persona mucha dependencia económica, el arreglo merece una revisión jurídica y humana, no solo una renegociación comercial. Si la dependencia es alta para ambos, la relación necesita procesos de voz, previsión y salida comparables a su materialidad. Si es baja para ambos, una interfaz ligera puede ser suficiente.

Esto no convierte todo compromiso en empleo ni toda concentración en abuso. Evita que la clasificación contractual borre la realidad económica. El *Authority-Dependency Map* del capítulo 12 comienza aquí: las propiedades de una relación no viajan siempre juntas. Puede haber poco control formal y mucha dependencia; mucha reserva y amplia autonomía; o un contrato estable con escasa memoria porque las personas rotan.

Una revisión bilateral debería producir dos balances. En el de la firma: calidad, latencia, coste total, conocimiento, continuidad y recovery. En el de la contraparte: previsibilidad, gastos, autonomía, concentración, aprendizaje, protección y capacidad de rechazar. El

objetivo no es maximizar todas las dimensiones. Es revelar el intercambio y evitar que una ganancia se presente como si nadie la financiara.

Control suficiente sin apropiación total

La distinción *controlled / committed / accessible* ayuda a descomponer lo que la palabra “control” mezcla. Una empresa puede necesitar control sobre el estándar, la aceptación y el dato, sin controlar el método diario del especialista. Puede necesitar un derecho de continuidad y exportación, sin poseer la herramienta. Puede conservar el juicio para detener una operación, mientras la ejecución pertenece a un partner.

Esta descomposición evita dos excesos. El primero es internalizar todo aquello que importa. Una capacidad puede ser crítica y estar bien gobernada mediante derechos, redundancias e interfaces externos. El segundo es creer que un contrato transfiere el problema completo. La empresa puede haber comprado ejecución y conservar, porque no puede o no debe delegarlos, los decision rights y la responsabilidad.

Para cada capacidad material conviene separar seis objetos: activo, conocimiento, decisión, ejecución, assurance y recuperación. Después se pregunta qué profundidad necesita cada uno. En seguridad, por ejemplo, una organización puede controlar políticas y respuesta a incidentes, mantener committed monitoring con un tercero y acceder a pruebas especializadas. En contenidos, puede controlar marca y aceptación, reservar una red recurrente y usar mercado spot para picos. No existe una posición única en el continuo porque la capacidad contiene piezas con propiedades distintas.

El mapa deja así de clasificar relaciones enteras y empieza a diseñar derechos. Esa es una mejora conceptual importante: no obliga a elegir entre “dentro” y “fuera” cuando el sistema real combina ambos.

La salida forma parte del compromiso

Una relación committed no está completa hasta que define cómo termina. Cuanto más contexto acumula, más fácil es posponer esta conversación. Sin embargo, memoria y dependencia crecen juntas. Una buena salida especifica qué artefactos se transfieren, cómo se revocan permisos, qué soporte de transición existe, qué información debe conservarse o eliminarse y quién responde por trabajos abiertos.

La exportabilidad no elimina todo switching cost. El conocimiento tácito no cabe entero en un repositorio. Pero una salida ensayable obliga a distinguir dependencia elegida de captura accidental. También mejora la relación presente: cuando datos, decisiones y responsabilidades están ordenados, la colaboración suele necesitar menos traducción personal.

El trigger de cambio debe ser conocido antes de la crisis. Puede ser una concentración excesiva, un *ramp time* que crece, un incidente, pérdida de continuidad, cambio de demanda o aumento de especificidad. Al alcanzarlo, la empresa no tiene solo dos opciones —renovar o romper—. Puede profundizar el compromiso, repartir ejecución, construir capacidad interna o reducir alcance.

El compromiso, por tanto, no es un contrato más largo. Es una arquitectura de reserva, memoria, derechos, reciprocidad y salida. Cuando esas piezas existen, la capa intermedia deja de ser ambigua y se convierte en una fuente deliberada de capacidad.

Las personas no se poseen. Las empresas controlan ciertos derechos y activos, comprometen ciertas relaciones y acceden a ciertas opciones.

NOTAS DEL CAPÍTULO

1. Sanford J. Grossman y Oliver D. Hart, “The Costs and Benefits of Ownership”, 1986; Oliver Hart y John Moore, “Property Rights and the Nature of the Firm”, 1990.
2. Unilever, “Preparing people to adapt and thrive in changing times”, 2021. Fuente corporativa; la vigencia y los resultados requieren verificación adicional.
3. Internal Revenue Service, Independent contractor or employee?.
4. Unión Europea, Directiva (UE) 2024/2831 relativa a la mejora de las condiciones laborales en el trabajo en plataformas.
5. Christopher Stanton y Catherine Thomas, “Who Benefits from Online Gig Economy Platforms?”, *American Economic Review*, 2025.
6. OIT, Digital labour platforms y *World Employment and Social Outlook 2021*.

HERRAMIENTA — CAPABILITY BOUNDARY AUDIT

Seleccione una capacidad, no una categoría de gasto. Complete la ficha con evidencia y fecha.

CAMPO	PREGUNTA DE AUDITORÍA
Outcome	¿Qué resultado y estándar debe producir?
Contractual boundary	¿Qué derechos y obligaciones existen hoy?
Accessible set	¿Qué fuentes son visibles en principio?
Usability gate	¿Cuáles superan availability, integration, assurance, learning, responsibility y recovery?
Layer	¿Qué está controlled, committed o accessible?
Total Access Cost	¿Qué precio, integración, revisión, error, memoria y salida implica?
Bilateral risk	¿Quién absorbe demanda, inversión, dependencia y fallo?
Trigger	¿Qué evidencia cambiaría la clasificación?

Output: un mapa, tres gaps y una decisión que todavía no puede tomarse. Si la auditoría produce solo una lista de proveedores, repítala: ha mapeado oferta, no capacidad.

Tres empresas que parecen iguales

Imagine tres compañías que producen el mismo servicio digital, facturan una cantidad comparable y declaran veinte empleados. En la primera, la mayor parte del producto y de la atención al cliente reside en la plantilla. En la segunda, diez personas coordinan una agencia de producto, varios especialistas, infraestructura cloud y un proveedor de soporte. En la tercera, el fundador y un pequeño equipo utilizan software e IA para automatizar gran parte del flujo, mientras proveedores externos resuelven excepciones.

En una tabla de headcount, las tres son iguales. En una narrativa superficial, la segunda y la tercera parecen más “líquidas” o más productivas. Ninguna conclusión resiste todavía. Hay que reconstruir el sistema.

La primera puede mantener más coste fijo y disponer de contexto, disponibilidad y aprendizaje acumulativo. También puede estar sobredimensionada, depender de conocimiento encerrado y tardar en incorporar especialización nueva. La segunda puede ajustar mejor la capacidad y acceder a expertise superior; también puede consumir muchas horas de integración y depender de una agencia que controla memoria. La tercera puede reducir ejecución rutinaria; también puede necesitar revisión intensa, cómputo, datos y un founder que resuelve cada excepción.

El **Capability Baseline** no decide cuál es mejor. Cambia las preguntas. Para un mismo outcome registra trabajo humano total, capital, software, inputs, calidad, latencia, aprendizaje y recuperación. La comparación deja de premiar una categoría

contractual.

La prueba de los tres denominadores

Con el denominador de empleo, la empresa con más output por persona parece superior. Con el denominador de trabajo humano total, parte de la diferencia puede desaparecer si el trabajo está en proveedores. Con un denominador económico completo, aparecen capital e inputs. Con el denominador de capacidad, la pregunta es si el sistema puede repetir y recuperarse.

No existe un único número correcto. Existe un número correcto para cada afirmación. Si se discuten obligaciones del empleador, payroll es central. Si se afirma productividad, se necesita valor añadido y el conjunto de inputs relevante. Si se afirma resiliencia, se necesita dependencia y recovery. Si se afirma capacidad futura, se necesita aprendizaje.

La regla editorial y directiva es la misma: una afirmación extraordinaria exige un denominador que no la favorezca por diseño. “Empresa de diez personas” describe empleo directo. “Diez personas producen todo” exige demostrar trabajo y capital totales. “La IA realizó el trabajo” exige incluir preparación, supervisión y fallo. “Accedemos al mejor talento” exige mostrar disponibilidad, integración y aceptación.

La prueba de desaparición

Retire una pieza material durante treinta días. En la primera empresa, puede faltar una persona clave. En la segunda, la agencia o el especialista committed. En la tercera, un proveedor de modelo, el dataset o quien revisa. ¿Qué outcome se degrada primero? ¿Quién diagnostica? ¿Qué permisos y artefactos permanecen? ¿Cuánto tarda

el reemplazo?

Esta prueba revela que control contractual y capacidad efectiva no coinciden. Una persona empleada puede ser insustituible porque el conocimiento no se distribuyó. Un proveedor externo puede ser recuperable gracias a estándares y portabilidad. Una herramienta puede parecer replicable y depender de configuraciones que nadie documentó.

La frontera útil no rodea a quienes “pertenecen”. Rodea la capacidad que puede movilizarse bajo un escenario y una fecha. Puede incluir relaciones estables y externas. Puede excluir activos formalmente controlados que ya no producen.

La prueba del aprendizaje

Después de diez ciclos, ¿qué empresa formula mejor el problema? La primera puede acumular experiencia en su plantilla, salvo que los departamentos no compartan. La segunda puede aprender mediante partners, salvo que cada deliverable se cierre sin transferencia. La tercera puede generar muchos datos, salvo que la automatización oculte razones y excepciones.

El aprendizaje no es una consecuencia automática de repetir. Necesita artefactos, feedback y una persona o relación que integre. La empresa que compra respuestas puede parecer eficiente y seguir siendo incapaz de juzgar. La que conserva cada paso dentro puede aprender y también reinventar lo que el mercado ya sabe.

La comparación correcta observa *learning location*: dónde mejora el modelo mental, quién puede reutilizarlo y qué ocurre al salir. No exige que toda fuente entregue su expertise. Exige que la empresa conserve aquello que necesita para decidir y responder.

La prueba bilateral

La segunda y la tercera empresa pueden ajustar su gasto a la demanda. ¿Quién absorbe la variación? Quizá un profesional reserva tiempo sin remuneración, una agencia mantiene bench, un proveedor concentra trabajo en otros clientes o usuarios realizan tareas que antes hacía la empresa. La flexibilidad no desaparece; cambia de balance.

La primera empresa asume más riesgo de demanda mediante empleo. A cambio puede obtener disponibilidad y compromiso, pero también trasladar presión interna mediante sobrecarga o despidos. Ninguna modalidad garantiza por sí sola un resultado humano mejor. El análisis debe observar ingresos, gastos, autonomía, protección, aprendizaje y voz de quienes contribuyen.

El objetivo no es producir un ranking moral. Es impedir que una arquitectura se llame eficiente porque uno de sus balances quedó fuera de la reunión.

El veredicto provisional

Las tres empresas pueden ser buenas organizaciones. La primera puede elegir empleo porque el bundle encaja. La segunda puede construir una red committed con interfaces y memoria. La tercera puede diseñar un workflow humano-IA con supervision y fallback. También pueden fracasar por rigidez, dependencia u opacidad.

La Parte I no demuestra que la frontera de payroll deba reducirse. Demuestra que no basta para describir capacidad. La ventaja, si existe, aparecerá en las decisiones que siguen: escoger modalidad, diseñar unidad y conservar aquello que el sistema necesita para aprender y responder.

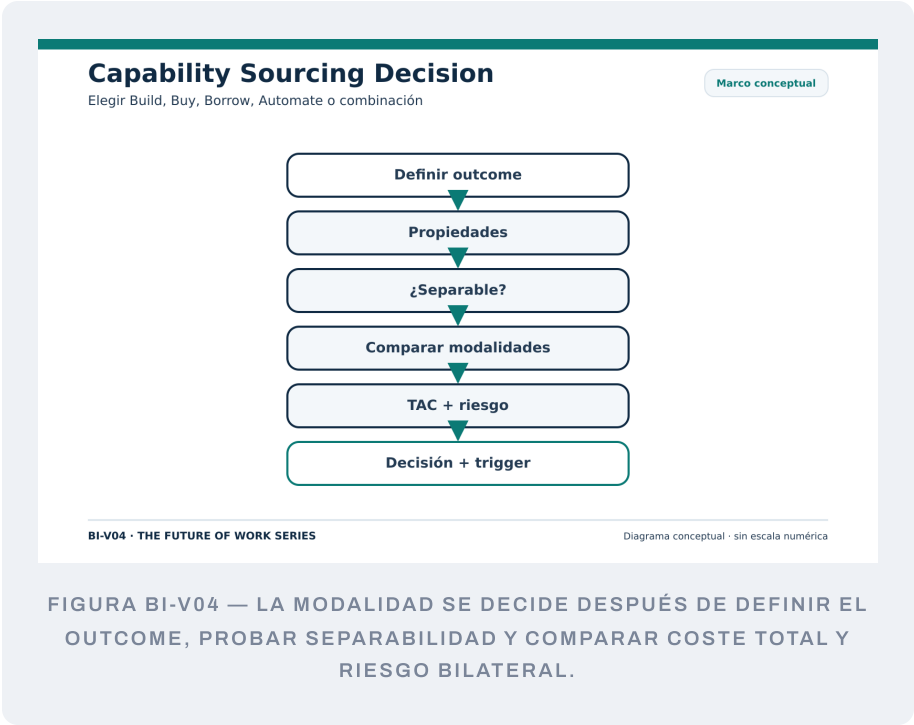
PARTE II

MÁS ALLÁ DE LA CONTRATACIÓN

Ver más capacidad no responde cómo obtenerla. Esta parte sustituye el reflejo de "abrir una vacante" por una elección contingente. Su objetivo no es externalizar más, sino escoger mejor y reconocer que un puesto es un bundle que a veces conviene conservar entero.

Capítulo 4

BUILD / BUY / BORROW / AUTOMATE



Elegir una modalidad sin enamorarse de ella

Pregunte cómo obtener una capacidad de ciberseguridad y recibirá cuatro respuestas seguras de sí mismas. “Forma al equipo.” “Contrata a un CISO.” “Llévalo a un proveedor especializado.” “Automatiza el monitoring.” Las cuatro pueden ser correctas. Las cuatro pueden ser peligrosas.

El error está en discutir modalidades antes de definir la capacidad. Cada recomendación convierte un instrumento familiar en punto de partida: learning para quien desarrolla talento, headcount para quien dirige un departamento, outsourcing para procurement, software para tecnología. El resultado puede parecer una decisión y ser solo un reflejo institucional.

La tesis más sólida de este libro es también la menos espectacular: **no existe una modalidad universalmente mejor**. La forma adecuada depende de las propiedades del trabajo, del mercado, del riesgo y de la institución que deberá integrarlo. Elegir no es identificar una casilla. Es diseñar una combinación y declarar qué evidencia obligará a revisarla.

La genealogía del menú

Laurence Capron y Will Mitchell popularizaron el marco *Build, Borrow, or Buy* para decisiones de recursos y crecimiento.¹ Esta obra lo extiende con **Automate**, no porque automatizar elimine a las otras modalidades, sino porque hoy forma parte del mismo problema de diseño. Antes de abrir una vacante o buscar un partner, una empresa puede preguntarse si reglas, software o IA ejecutarán ciertos pasos.

Los cuatro verbos necesitan precisión:

- **Build:** desarrollar internamente habilidad, rutina, conocimiento o sistema.
- **Buy:** obtener control estable, mediante empleo, activo, tecnología o adquisición. En personas, *buy* nunca significa propiedad.
- **Borrow:** acceder a capacidad mediante independiente, firma, alianza, comunidad o acuerdo de reserva.
- **Automate:** delegar pasos a reglas, software o IA, con criterios, supervisión y fallback.

La respuesta suele ser híbrida. Una empresa puede construir criterio interno, contratar a quien conserva decision rights, usar un proveedor para ejecución especializada y automatizar comprobaciones repetitivas. El valor del framework está en hacer explícita la arquitectura, no en forzar una única letra.

Paso 1: outcome antes que modalidad

“Necesitamos datos” no define una capacidad. “Necesitamos detectar fraude” tampoco basta. Un outcome utilizable añade estándar y contexto: detectar determinadas operaciones antes de autorización, con una tasa aceptable de falsos positivos, trazabilidad, escalado humano y cumplimiento aplicable.

La definición debe incluir:

- resultado y usuario;
- frecuencia y volumen;
- latencia tolerable;
- umbral de calidad;
- errores severos y daño;
- continuidad requerida;
- aprendizaje que conviene conservar;
- accountable owner.

Sin esta especificación, comparar modalidades equivale a comparar precios de objetos distintos.

Paso 2: propiedades de la capacidad

La transaction cost economics de Oliver Williamson propone mirar especificidad, incertidumbre y frecuencia, entre otras dimensiones.²

La literatura posterior añade conocimiento, complementariedades e incentivos. Para una decisión práctica conviene preguntar:

Especificidad. ¿Cuánto debe invertir la parte en activos, conocimiento o relaciones que valen menos fuera de esta empresa? Una capacidad muy específica puede necesitar mayor compromiso o salvaguardas.

Conocimiento tácito. ¿Puede explicarse con documentación o requiere experiencia compartida, observación y criterio acumulado?

Frecuencia y volatilidad. ¿La necesidad es continua, intermitente, estacional o impredecible? La media puede ocultar picos críticos.

Interdependencia. ¿El output tiene una interfaz estable o cambia

con muchas piezas? A mayor interdependencia, mayor impuesto de coordinación.

Criticidad y daño. ¿Qué ocurre si falla? ¿La calidad puede verificarse antes de causar daño? ¿Quién responde?

Aprendizaje. ¿Cada ejecución mejora una capacidad estratégica o produce conocimiento que otros pueden apropiarse?

Profundidad de mercado. ¿Existen alternativas comparables o un proveedor dominante? ¿Puede sustituirse la fuente sin reconstruir meses de contexto?

Ninguna propiedad decide sola. Una tarea frecuente puede automatizarse; si su daño potencial es alto, necesitará controles. Una capacidad rara puede pedirse prestada; si define la ventaja, quizá la empresa deba construir absorptive capacity dentro.

Paso 3: probar la separabilidad

Antes de elegir quién ejecuta, pregunte cuál es la unidad. Una actividad puede parecer externalizable hasta que se dibujan dependencias, excepciones y aprendizaje. El **Separability Test**, que desarrollaremos en el capítulo 6, evita comprar un fragmento cuyo valor depende de un bundle que nadie reconstruirá.

Este paso es especialmente importante con IA. Una demo puede automatizar una tarea y dejar intacto —o aumentar— el trabajo de preparación, revisión y escalado. La unidad no es “generar respuesta”; es resolver el caso con calidad y responsabilidad.

Paso 4: determinar el mínimo de control

La empresa no necesita controlar todo. Necesita controlar lo

suficiente. El **Solid Minimum Test** pregunta si perder una capacidad amenazaría ventaja, aprendizaje, seguridad, cumplimiento, recuperación, confianza o decision rights. Si la respuesta es sí, *Borrow* todavía puede formar parte de la solución, pero debe existir un núcleo estable que gobierne la dependencia.

Los debates públicos sobre insourcing ofrecen un ejemplo limpio del mecanismo. La Government Accountability Office de Estados Unidos ha descrito cómo las agencias deben equilibrar empleados y contractors para conservar control sobre funciones críticas y misiones públicas.³ No toda función necesita realizarse internamente; ciertas capacidades, sin embargo, no pueden volverse tan externas que la institución deje de dirigirlas.

Paso 5: comparar coste total y riesgo bilateral

Cada modalidad tiene costes que sus defensores omiten.

Build consume tiempo y puede reinventar una capacidad disponible. **Buy** crea compromiso fijo y riesgo de rigidez. **Borrow** añade búsqueda, integración, dependencia y switching. **Automate** requiere datos, pruebas, permisos, monitoring, review y recovery.

Compare:

- coste directo;
- time-to-capability;
- coste de integración y coordinación;
- calidad y variabilidad;
- conocimiento retenido;
- dependencia y concentración;
- coste de error y recuperación;
- reversibilidad;
- riesgo y bienestar de las personas afectadas.

La última línea impide que la flexibilidad de la empresa se calcule trasladando toda variación hacia alguien con menos poder. La decisión puede seguir siendo externa, pero deja de fingir que el coste desapareció.

Paso 6: diseñar una combinación

P&G Connect + Develop no sustituye I+D interna por una multitud indiferenciada. Su descripción pública muestra búsqueda externa y construcción de partnerships.⁴ El mecanismo útil es la combinación: la empresa amplía el origen de ideas y conserva capacidades para formular, seleccionar, integrar y llevar al mercado.

En ciberseguridad, una arquitectura podría ser:

- principios, risk appetite y incident ownership bajo control estable;
- capacidad especializada de threat hunting comprometida con un partner;
- pruebas puntuales accesibles;
- monitoring automatizado con escalado humano;
- conocimiento de incidentes retenido dentro.

Observe que no hemos “elegido proveedor”. Hemos distribuido una capacidad por propiedades. La unidad que debe responder sigue completa.

Paso 7: preservar una opción y definir el trigger

La decisión más inteligente puede ser la que conserva la posibilidad de cambiar. Un piloto, un acuerdo con exportabilidad, una arquitectura modular o una inversión mínima interna pueden reducir el coste de reversión. Esta opción tiene valor cuando la demanda o la tecnología son inciertas.

Cada decisión debe registrar un **trigger**:

- si el volumen supera X durante Y meses, reevaluar compromiso estable;
- si el time-to-ramp excede el umbral, simplificar interfaz o internalizar;
- si la tasa de incidentes alcanza un nivel, bloquear automatización;
- si la concentración de proveedor supera un límite, crear alternativa;
- si el conocimiento no se transfiere, cambiar el contrato o la modalidad.

El trigger evita evaluar la decisión por orgullo. Internalizar después de externalizar no es admitir una derrota; puede ser la respuesta correcta a propiedades que cambiaron o que la experiencia reveló.

Boeing y la responsabilidad que no viaja

La cadena de aviación ofrece un límite potente a la narrativa asset-light. Un informe de GAO sobre fabricación de aviación comercial documentó dificultades para elevar producción tras 2020, incluidas escasez de workforce y materiales en Boeing, Airbus y sus proveedores.⁵ La enseñanza no es “outsourcing causa escasez”. Sería una inferencia demasiado fuerte. Es más sobria: una red extensa no elimina dependencia, visibilidad, calidad ni recuperación. Cuando el sistema falla, el fabricante no puede responder “la capacidad estaba fuera”.

Externalizar ejecución puede cambiar quién hace el trabajo. No cambia automáticamente quién responde ante cliente, regulador o público. Cuanto más crítica sea la capacidad, más importante es especificar control, información, auditabilidad y fallback.

Capability Sourcing Decision Record

El BBBA Decision System termina en un registro breve, fechado y refutable:

1. outcome y umbral;
2. unidad tras Separability Test;
3. Solid Minimum requerido;
4. alternativas BBBA y combinación elegida;
5. coste total y riesgo bilateral;
6. opción preservada;
7. trigger, métricas y fecha de revisión.

Registrar ex ante importa porque la contingencia tiene un defecto: puede justificar cualquier cosa a posteriori. Si el resultado es bueno, declaramos que la modalidad encajaba. Si es malo, decimos que la ejecución falló. Un record obliga a declarar la hipótesis, la alternativa y la señal de error antes de conocer el desenlace.

La decisión no termina al firmar

Una contratación, un contrato o una automatización no “adquieren” capacidad por sí mismos. Abren un proceso de integración. La modalidad puede ser correcta y fallar por una interfaz débil; puede ser imperfecta y funcionar gracias a una relación excelente. Separar elección y ejecución nos ayuda a aprender.

El principal cambio directivo es sencillo: no apruebe un *hire*, un vendor o una herramienta sin saber qué propiedad de la capacidad lo exige y qué outcome haría cambiar de opinión.

Elegir modalidad obliga ahora a elegir la unidad. A veces la

empresa no necesita un puesto completo. Quizá necesita una fracción recurrente de criterio. Pero el tiempo puede dividirse con más facilidad que la responsabilidad. **¿Cuándo funciona realmente una capacidad fraccional?**

Las modalidades cambian aquello que la empresa aprenderá

Build, Buy, Borrow y Automate no son solo maneras de conseguir el mismo output. Cada una produce una trayectoria distinta de conocimiento. **Build** obliga a descubrir y codificar; puede crear una base propia y también consumir tiempo en problemas ya resueltos fuera. **Buy**, entendido como incorporar capacidad estable, facilita aprendizaje acumulativo y puede encerrar el sistema en una estructura difícil de redimensionar. **Borrow** trae experiencia y variedad; puede acelerar la solución y dejar el razonamiento fuera. **Automate** convierte parte de la práctica en reglas, datos o modelos; puede hacer el proceso más repetible y degradar la capacidad humana de detectar excepciones.

La decisión debería responder dos preguntas temporales: ¿qué necesitamos producir ahora?, y ¿qué queremos ser capaces de producir después de haberlo hecho varias veces? Una modalidad puede ganar la primera comparación y perder la segunda. Contratar una firma experta para lanzar una operación puede ser correcto si el objetivo incluye transferir prácticas. Puede ser una dependencia si cada ciclo vuelve a comprar el mismo diagnóstico. Desarrollar dentro puede parecer lento y, sin embargo, construir el conocimiento que permitirá evaluar proveedores más adelante.

El **learning path** debe formar parte del Decision Record. Para cada alternativa se declara qué aprenderá la organización, en qué artefactos quedará, quién podrá aplicarlo y qué conocimiento

seguirá perteneciendo a la contraparte. No todo debe transferirse. La empresa paga legítimamente por expertise que no necesita poseer. Lo que no puede hacer es descubrir, demasiado tarde, que externalizó la capacidad de juzgar aquello de lo que sigue respondiendo.

Elegir una secuencia, no una casilla

Las propiedades de una capacidad cambian. Una necesidad incierta puede empezar como *Borrow* para explorar, pasar a *Build* cuando aparece aprendizaje estratégico, incorporar *Buy* para dar continuidad y automatizar pasos maduros. Otra capacidad puede seguir el camino inverso: una unidad interna se convierte en servicio compartido, se estandariza y después se contrata externamente con una interfaz estable.

Pensar en secuencias reduce el dogmatismo. La pregunta no es cuál modalidad representa el futuro, sino qué modalidad compra la opción adecuada en esta fase. Un piloto externo puede producir información sobre volumen y separabilidad antes de asumir coste fijo. Un equipo interno pequeño puede conservar criterio mientras el mercado demuestra profundidad. Una automatización parcial puede generar datos de errores antes de ampliar autonomía.

La secuencia necesita hitos. En una capacidad nueva, el primer hito puede ser demostrar el outcome. El segundo, estabilizar el workflow. El tercero, medir demanda y excepciones. El cuarto, decidir dónde residirá el aprendizaje. Sin hitos, el piloto se vuelve permanente por inercia o la contratación se justifica con necesidades que nunca se midieron.

También importa el orden dentro de una combinación. Automatizar antes de clarificar el proceso puede congelar

ambigüedad. Externalizar antes de definir assurance puede dar al proveedor la capacidad de evaluarse. Contratar antes de resolver datos y permisos puede incorporar talento excelente a un sistema que no le deja producir. BBBA es, por ello, una arquitectura de intervención: primero outcome y unidad; después control, interfaz, modalidad y aprendizaje.

El contrafactual que falta

Las decisiones de sourcing suelen evaluarse sin alternativa explícita. Si un proveedor entrega, se declara éxito; si un equipo interno cumple, se justifica su existencia. Pero ninguna observación muestra por sí sola que la modalidad fue la mejor disponible. El Decision Record debe conservar el contrafactual razonable: qué habría ocurrido con la segunda mejor combinación, bajo qué supuestos y con qué coste de cambio.

No se necesita un experimento perfecto. Basta con evitar la evaluación circular. Si la razón para *Borrow* fue velocidad, mida acceso y *ramp time* frente a la alternativa interna. Si la razón para *Buy* fue aprendizaje, compruebe si decisiones y performance mejoran con la acumulación. Si la razón para *Automate* fue consistencia, mida dispersión y errores severos además de tiempo. Si la razón para *Build* fue diferenciación, observe si el conocimiento resultante afecta outcomes que el mercado no ofrece.

La ausencia de contrafactual favorece a la modalidad más visible. El coste de una persona contratada aparece en cada mes; el coste de coordinación externa puede quedar repartido entre managers. El tiempo de desarrollo interno se registra; el tiempo de revisión de una herramienta puede diluirse en la jornada. Hacer explícita la alternativa no elimina el sesgo, pero fuerza un denominador simétrico.

Reversión como señal de aprendizaje

Insourcing, cambio de proveedor o retirada de automatización suelen narrarse como fracaso. En un sistema contingente pueden ser la respuesta correcta. La frontera se mueve cuando cambian volumen, especificidad, regulación, market depth, tecnología o coste de error. También se mueve cuando la primera decisión revela que nuestras hipótesis eran falsas.

Una reversión bien gobernada conserva tres cosas: el motivo original, la evidencia que lo invalidó y el conocimiento generado durante la modalidad anterior. Sin ese registro, la organización oscila entre modas. Externaliza porque “hay que ser ligera”, internaliza tras un incidente, automatiza al siguiente ciclo y vuelve a contratar cuando falla. Cada movimiento parece estratégico y ninguno acumula aprendizaje.

El trigger reduce esa oscilación. Debe ser suficientemente específico para obligar a revisar y suficientemente prudente para no automatizar una decisión compleja. “Revisar si el proveedor no funciona” es inútil. “Revisar si el *ramp time* supera cuatro semanas durante dos proyectos, si el retrabajo excede el umbral o si ningún responsable interno puede aceptar el output sin ayuda del proveedor” crea evidencia. Los valores concretos dependen del caso; la estructura no.

El límite de Automate

Añadir Automate al marco evita que toda necesidad se traduzca en trabajo humano. También puede crear la falsa impresión de que la tecnología es una modalidad sin contraparte. La automatización depende de personas que diseñan, proporcionan datos, operan infraestructura, revisan y responden. Su coste se desplaza desde

ejecución visible hacia ingeniería, assurance y recovery.

La pregunta correcta no es si una herramienta “hace el trabajo”, sino qué porcentaje del workflow ejecuta bajo qué condiciones. Un sistema puede redactar una primera versión y no decidir el estándar. Puede clasificar casos rutinarios y escalar excepciones. Puede actuar en sistemas con permisos limitados y necesitar confirmación para acciones irreversibles. Cada paso automatizado cambia la distribución de atención humana; no necesariamente la elimina.

Automate debe compararse con Build, Buy y Borrow usando el mismo objeto: outcome end-to-end. Si el ahorro de generación se consume en revisión, si los errores severos aumentan o si nadie puede recuperar el proceso sin el proveedor tecnológico, la capacidad no se ha vuelto más barata en el sentido relevante. Puede seguir siendo la mejor opción, pero la decisión debe reconocer el sistema completo.

Del sourcing a la arquitectura

El resultado maduro del framework no es una letra, sino un diagrama de responsabilidades. Muestra qué se desarrolla, qué se incorpora de forma estable, qué se reserva externamente y qué pasos ejecuta tecnología; coloca encima los decision rights, el assurance y la memoria; y añade un trigger. Ese diagrama permite que HR, procurement, tecnología, operaciones y riesgo discutan el mismo outcome.

Esta conversación transversal es parte del valor. Muchas malas decisiones no proceden de elegir el proveedor equivocado, sino de que cada función optimiza una parte distinta. HR mira puestos, procurement tarifas, tecnología integraciones, finanzas gasto y operaciones velocidad. BBBA crea un objeto común y obliga a

reconciliarlo antes de comprometer la modalidad.

La conclusión más exigente del framework es que ninguna modalidad exige de diseñar. Acceder no evita integrar. Contratar no evita especificar. Desarrollar no evita comparar. Automatizar no evita responder. La empresa no compra capacidad terminada: construye una configuración capaz de producirla.

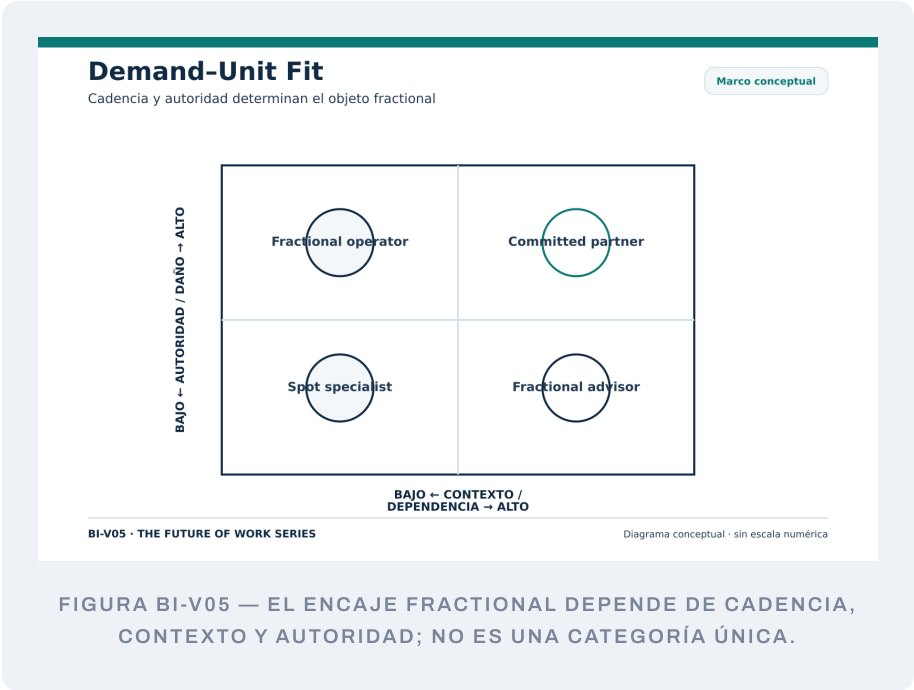
No elija entre interno y externo. Diseñe una combinación, declare su coste total y escriba la señal que obligará a revertirla.

NOTAS DEL CAPÍTULO

1. Laurence Capron y Will Mitchell, *Build, Borrow, or Buy*, Harvard Business Review Press, 2012.
2. Oliver E. Williamson, “*Transaction Cost Economics: The Natural Progression*”, Nobel Lecture, 2009.
3. U.S. GAO, *Civilian Agencies’ Development and Implementation of Insourcing Guidelines*, 2009, y *Sourcing Policy*, 2010.
4. Procter & Gamble, *Innovation — Connect + Develop*.
5. U.S. GAO, *Commercial Aviation Manufacturing: Supply Chain Challenges and Actions to Address Them*, 2024.

Capítulo 5

FRACTIONAL EVERYTHING?



La unidad mínima de capacidad

Imagine un calendario, no una empresa concreta. El lunes aparece una decisión de financiación. El miércoles, una anomalía en el cierre. El viernes por la noche, una pregunta del consejo. En el calendario hay un CFO fraccional reservado para el martes y el jueves por la mañana. ¿Qué ha comprado la empresa: dos bloques de tiempo, criterio disponible durante la semana o responsabilidad financiera continua?

La pregunta explica casi toda la promesa y casi todo el riesgo de lo fractional. Una necesidad puede ocupar solo una parte de un puesto. La responsabilidad que la rodea no siempre respeta la misma fracción.

El lenguaje se ha expandido desde fractional CFO hasta CTO, CMO, CHRO, CISO, product leader, general counsel y operaciones. La oferta existe. Lo que todavía no existe con la misma solidez es evidencia independiente que permita afirmar una superioridad general. Gran parte de los casos disponibles procede de intermediarios o de quienes venden el servicio. Por eso el signo de interrogación del título es metodológico.

Antes de la etiqueta

Lo fractional no nació en redes sociales. Interim managers, asesores, partners de firmas profesionales y carreras portfolio llevan décadas ofreciendo unidades no permanentes de dirección o expertise. La novedad relativa está en la facilidad de descubrir oferta, la normalización de la etiqueta y la posibilidad de servir a empresas más pequeñas de forma recurrente y remota.

La etiqueta agrupa objetos distintos. Una compañía puede comprar:

- horas de ejecución;
- una cadencia de revisión;
- criterio ante decisiones;
- una firma o función regulada;
- coaching para un equipo;
- acceso a una red;
- diseño de sistema;
- responsabilidad por un outcome.

Dos días al mes pueden ser suficientes para revisar una arquitectura y absurdos para responder a incidentes. Un consejo estratégico puede tolerar latencia; una obligación regulatoria quizá no. Antes de preguntar qué fracción contratar, hay que identificar qué unidad de capacidad se necesita.

La curva de ajuste

El argumento económico es intuitivo. Si una empresa necesita veinte horas mensuales de experiencia senior, contratar un puesto completo puede producir coste fijo y una mala correspondencia entre necesidad y unidad. Un profesional fractional puede servir esa demanda y distribuir su capacidad entre varias organizaciones.

Pero la necesidad rara vez llega como una línea plana. Tiene picos, excepciones y preparación. Las veinte horas facturables pueden exigir tiempo no visible para recordar contexto, coordinar y mantenerse actualizado. Una crisis puede concentrar la necesidad justo cuando otros clientes también la tienen. La fracción que encaja en la media puede fallar en la cola de la distribución.

El ajuste depende de tres curvas:

1. **volumen:** cuánta capacidad se necesita;
2. **cadencia:** con qué regularidad y anticipación;
3. **criticidad temporal:** qué daño causa no estar disponible fuera de la agenda.

Una necesidad parcial, previsible y delimitable es el terreno más favorable. Una crisis permanente, autoridad diaria o conocimiento tácito dominante sugieren mayor compromiso o control.

Fractional Fit Test

El **Fractional Fit Test** no produce una nota. Obliga a contestar nueve preguntas.

1. VOLUMEN Y CADENCIA

¿La demanda cabe de forma estable en una fracción o alterna entre ociosidad y desbordamiento? ¿Puede planificarse?

2. MODULARIDAD

¿Existe un outcome y una interfaz definidos? ¿La persona puede producir sin reconstruir cada vez el trabajo de otros?

3. CONTEXTO

¿Qué proporción de la fracción se consumirá entendiendo lo ocurrido en su ausencia? ¿Cómo se mantiene la memoria?

4. DISPONIBILIDAD

¿Qué está reservado? ¿Existe respuesta fuera de cadencia? ¿Quién cubre vacaciones, conflicto o emergencia?

5. AUTORIDAD

¿Puede decidir o solo recomendar? ¿Quién asigna recursos y ejecuta?
Un título C-level sin decision rights puede crear una ficción.

6. CONFLICTO

¿Hay clientes competidores, calendarios correlacionados o deberes incompatibles? ¿Cómo se declaran y resuelven?

7. REGULACIÓN

¿La función exige presencia, independencia, firma, supervisión o estatus específico? La respuesta cambia por jurisdicción.

8. APRENDIZAJE

¿La relación desarrolla al equipo y transfiere sistema, o crea dependencia personal? ¿Qué ocurre al terminar?

9. OWNER END-TO-END

¿Quién responde cuando el profesional no está? La empresa no puede fraccionar su deber hacia clientes, inversores o reguladores.

Las salidas son cuatro: **fit**, **fit condicionado**, **committed capacity** o **control estable recomendado**. No hay salida “fractional porque es más barato”.

Committed, no spot

Las mejores condiciones fractional suelen parecerse más a la capa committed que al mercado spot. Hay reserva, tenure, contexto, una cadencia y expectativas de escalado. La relación puede ofrecer lo que una consulta puntual no ofrece: memoria.

Esa memoria debe diseñarse. Un decision log, un mapa de riesgos,

una persona interna capaz de continuar y una obligación de transferencia reducen dependencia. También permiten que la relación evolucione: la necesidad puede disminuir porque el sistema mejora, o aumentar hasta justificar una función estable.

La empresa debería querer que el fractional professional pueda hacer su presencia menos necesaria en ciertas tareas, incluso si conserva un rol de criterio. De otro modo, ha comprado una dependencia con tarifa flexible.

Founder-orchestration debt

Una organización joven puede reunir un CFO fractional, una agencia de marketing, un estudio de producto, un asesor legal y varios contractors. Sobre el papel dispone de un equipo ejecutivo ampliado. En la práctica, alguien debe mantener prioridades coherentes, resolver interfaces y decidir cuando las agendas no coinciden. Con frecuencia esa persona es el founder.

La suma de fracciones crea **founder-orchestration debt** cuando la integración vuelve a una sola cabeza:

- cada especialista recibe una versión distinta del contexto;
- nadie posee el workflow completo;
- decisiones se retrasan hasta una reunión conjunta;
- el founder traduce entre vocabularios;
- el conocimiento queda repartido por documentos y bandejas;
- una ausencia hace visible la falta de sistema.

Lo fractional reduce capacidad ociosa en cada función y puede aumentar el coste de coordinación entre funciones. El ahorro no se evalúa por línea, sino por outcome.

Bilateral Performance

La empresa no es la única que necesita un buen fit. El profesional gestiona un portfolio. Su performance depende de tiempo no facturable, previsibilidad, concentración de ingresos, posibilidad de decir no, inversión formativa y conflicto entre demandas.

Una arquitectura que presupone disponibilidad instantánea sin pagar reserva no ha comprado flexibilidad; ha imaginado capacidad gratuita.

La investigación sobre cómo quienes trabajan por cuenta propia perciben costes y beneficios muestra heterogeneidad: control y flexibilidad pueden ser valiosos, mientras gastos, incertidumbre y ausencia de protecciones importan.¹ No existe una experiencia freelance única.

Un scorecard bilateral debería preguntar:

- ¿la empresa recibe criterio y continuidad suficientes?
- ¿el profesional puede sostener calidad sin sobrecompromiso?
- ¿ambas partes entienden tiempo reservado y no reservado?
- ¿el precio cubre preparación y responsabilidad?
- ¿existe voz sobre scope, evaluación y cambios?

La relación solo es elástica si no depende de que una de las partes absorba silenciosamente toda variación.

La objeción incómoda

“Fractional” puede ser una forma elegante de decir que una empresa no puede financiar la función. También puede ser precarización premium: seniority y marca profesional ocultando inseguridad o

trabajo no pagado. Es una objeción real.

La respuesta no consiste en defender la etiqueta. Consiste en exigir evidencia de elección y unit economics. ¿La unidad parcial corresponde a una necesidad parcial? ¿Existe una alternativa estable y por qué no se elige? ¿Quién absorbe riesgo? ¿El precio cubre la modalidad? ¿Hay independencia real? Si las respuestas no sostienen el diseño, la empresa necesita otra cosa.

También existe el error contrario. Una organización puede contratar prematuramente un puesto senior completo, llenar su tiempo con tareas de menor valor y crear una jerarquía antes de necesitarla. El empleo no es automáticamente más responsable; depende del outcome y del bundle.

Lo que sabemos y lo que todavía no

Sabemos que la capacidad fraccional existe y que algunas necesidades intermitentes admiten una unidad menor que un puesto. Sabemos, por teoría de contratos y trabajo, que reserva, contexto y control cambian el arreglo. No sabemos con evidencia comparativa suficiente qué modalidades fraccional mejoran outcomes en distintas funciones, cuánto tiempo de coordinación ocultan ni qué carreras producen a largo plazo.

Esa incertidumbre no inutiliza el concepto. Define su claim ceiling. No concluya “fractional everything”. Pregunte “fractional what, when, with what owner and at whose risk?”.

Fraccionar tiempo puede mejorar el ajuste. Pero un puesto contiene más que tiempo: relaciones, excepciones, mentoring, identidad y derechos. El siguiente capítulo seguirá ese bundle para descubrir

qué se rompe cuando desagregamos un job y qué debemos reconstruir.

Cuatro objetos distintos bajo la misma etiqueta

La conversación sobre *fractional leadership* mejora cuando se distinguen cuatro objetos que el mercado suele presentar juntos. El primero es el **advisor**: aporta criterio en momentos definidos, pero no ejecuta ni posee el outcome. El segundo es el **operator**: ejecuta una parte recurrente del workflow con un alcance delimitado. El tercero es el **system builder**: diseña procesos, desarrolla a un equipo y deja una capacidad que otros podrán operar. El cuarto es el **accountable executive**: posee decisiones, asigna recursos y responde de forma continua por una función.

Los cuatro pueden utilizar un número parecido de horas y producir relaciones radicalmente distintas. El advisor funciona si la organización tiene capacidad para convertir recomendación en decisión. El operator necesita una interfaz y un owner end-to-end. El system builder exige transferencia, hitos y una definición de “terminado”. El accountable executive necesita autoridad y disponibilidad compatibles con la responsabilidad; fraccionar su agenda sin fraccionar las consecuencias puede crear un título sin institución.

Antes de contratar, la empresa debería escribir qué objeto necesita. Si pide “un CFO fractional” y espera asesoramiento de financiación, cierre mensual, tesorería, relación con el consejo, compliance y respuesta a cualquier incidencia, no ha definido una fracción; ha comprimido una función. La respuesta puede ser combinar un responsable interno, un especialista recurrente y proveedores operativos. También puede ser contratar. El test evita que la etiqueta reemplace al diseño.

La cola de excepciones

La demanda media favorece lo fractional; la cola de excepciones decide si funciona. Una función puede requerir ocho horas semanales en condiciones normales y cuarenta durante una financiación, una inspección, un incidente o una reestructuración. Si varios clientes atraviesan el mismo shock —cierre fiscal, cambio regulatorio o crisis de mercado— la disponibilidad del profesional se vuelve correlacionada.

El contrato debe tratar esa cola. Puede reservar capacidad adicional, identificar un equipo de respaldo, definir qué eventos activan otra modalidad o mantener un owner interno capaz de sostener la operación. Fingir que la fracción regular cubre cualquier contingencia crea una promesa imposible. Pagar toda la cola permanentemente también puede ser ineficiente. El objetivo es distribuirla con intención.

Una herramienta útil es separar tres niveles de servicio. **Cadencia:** trabajo previsto que cabe en la fracción. **Escalado:** respuesta extraordinaria con condiciones, coste y límites explícitos. **Crisis:** evento que excede la relación y activa un plan diferente. Esta distinción protege al profesional de una disponibilidad implícita y a la empresa de confundir acceso con cobertura.

El ciclo de vida de una relación fractional

La capacidad fraccional no debería evaluarse solo en el momento de entrada. Tiene un ciclo. Al principio compra velocidad y experiencia mientras la organización todavía no conoce la forma final de la función. Después puede estabilizar prácticas y formar a un equipo. Más tarde, la necesidad puede seguir parcial, migrar a una relación más comprometida, convertirse en empleo o desaparecer porque el

sistema ya funciona.

Cada fase necesita un outcome diferente. En la fase de diagnóstico: mapa de decisiones, riesgos y backlog. En la de construcción: procesos, datos, interfaces y responsables. En la de transferencia: evidencia de que otras personas pueden operar. En la de continuidad: una cadencia de revisión que no reconstruye el contexto. En la de salida: memoria, permisos y obligaciones cerrados.

Sin esta secuencia, el arreglo puede quedar atrapado. La persona es demasiado importante para terminar y demasiado externa para construir un equipo autónomo. Las tareas operativas ocupan el tiempo de criterio. La empresa sigue diciendo que compra experiencia senior y utiliza a ese profesional como un sistema manual de integración. El coste no está en la tarifa, sino en una función que nunca madura.

Por eso el contrato de inicio debería contener una hipótesis de evolución. No es una fecha de caducidad automática. Es una respuesta a “¿qué sería distinto dentro de doce meses si esta relación funciona?”. Puede ser mejor información, un equipo desarrollado, un proceso más estable, una decisión de contratar o la confirmación de que la fracción es la unidad correcta. Si nada debería cambiar, la empresa debe justificar por qué esa dependencia permanente es deseable.

Autoridad, presencia y legitimidad

Un título ejecutivo produce expectativas dentro y fuera. Un CMO puede representar la función ante el equipo; un CFO puede interactuar con consejo, bancos o auditores; un CHRO puede decidir sobre políticas que afectan a personas. La legitimidad no procede

solo del expertise. También requiere claridad sobre presencia, autoridad, conflictos y acceso a información.

Los equipos necesitan saber cuándo una recomendación es una decisión, quién puede comprometer presupuesto y quién resolverá un desacuerdo. Si el profesional tiene título pero no derechos, las decisiones regresan al founder. Si tiene derechos amplios y supervisión diaria, la independencia contractual puede no reflejar la relación real. El diseño debe encontrar una forma coherente, no una etiqueta comercial atractiva.

La legitimidad se construye con rituales concretos: mandato público, decision log, cadencia accesible, criterios de escalado y una persona interna que mantiene continuidad. También con límites: conflictos declarados, datos segregados, autorización para representar a la empresa y reglas sobre información de otros clientes. El profesional no debe convertirse en un canal informal entre organizaciones.

La perspectiva profesional

Una carrera portfolio puede ofrecer autonomía, variedad y aprendizaje acelerado. También contiene trabajo invisible: desarrollo comercial, preparación, facturación, herramientas, seguro, formación y huecos entre encargos. La empresa que compara una tarifa con un salario debe recordar que las unidades incorporan distribuciones de coste distintas. No corresponde convertir esa realidad en una regla moral única; corresponde reflejarla en precio, reserva y expectativas.

La calidad del portfolio depende de concentración y compatibilidad. Un profesional con diez clientes pequeños puede sufrir más fragmentación que otro con tres relaciones profundas. Un

calendario lleno puede reducir la capacidad de pensar. Varios clientes del mismo sector pueden crear conflictos; varios sectores pueden multiplicar el coste de contexto. La arquitectura bilateral debe permitir que la contraparte rechace, renegocie y proteja la calidad.

Medir *Bilateral Performance* significa preguntar si el arreglo puede sostenerse sin que una parte subvencione a la otra con disponibilidad gratuita, riesgo excesivo o trabajo no visible. No exige igualdad perfecta. Exige que el intercambio deje de esconderse bajo la palabra flexibilidad.

Cuándo no utilizarlo

Hay cinco señales fuertes contra una solución fractional. La primera es una demanda cotidiana que ya llena una función. La segunda, una autoridad que debe estar disponible de forma continua. La tercera, un aprendizaje estratégico que se acumula con cada decisión y que la empresa no sabe retener. La cuarta, una interdependencia que convierte cada ausencia en rebrief. La quinta, una relación de control y dependencia que no encaja con independencia real.

También hay señales contra contratar: necesidad experimental, volumen bajo, mercado de expertise profundo, capacidad interna suficiente para integrar y alto valor de reversibilidad. El propósito del test es permitir ambas conclusiones.

La promesa útil de *Fractional Everything* no es que cada puesto pueda cortarse en porcentajes. Es que la empresa puede dejar de confundir necesidad parcial con puesto completo. Su límite es igual de importante: dividir tiempo no divide automáticamente autoridad, aprendizaje ni responsabilidad.

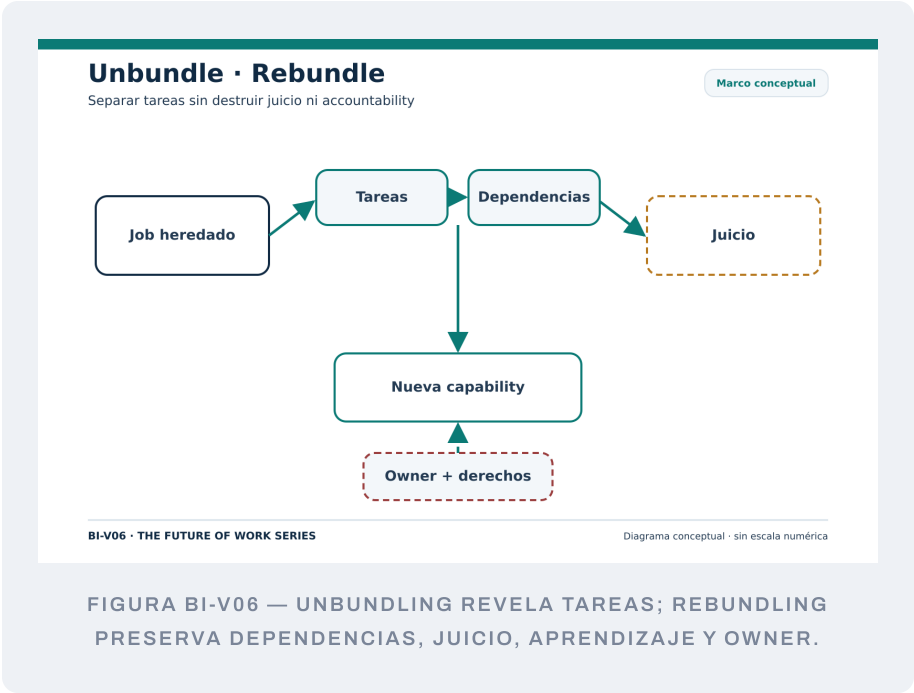
Especifique primero la fracción de necesidad y después la fracción de tiempo. La responsabilidad end-to-end permanece entera.

NOTAS DEL CAPÍTULO

1. Edward Freeland, Andrew Garin y Dmitri Koustas, “How Do Workers Think About the Costs and Benefits of Freelance Work?”, NBER, 2026.

Capítulo 6

THE UNBUNDLING—AND REBUNDLING—OF THE JOB



Lo que el puesto mantenía unido

Un puesto parece distinto cuando lo leemos en una descripción y cuando lo observamos durante una semana. La descripción enumera tareas: responder consultas, preparar informes, revisar contratos, escribir código. La semana real añade interrupciones, excepciones, conversaciones, mentoring, prioridades cambiantes y decisiones que nadie había escrito.

La digitalización vuelve separables más pasos. Las plataformas permiten encargar una pieza. El trabajo remoto reduce la necesidad de copresencia. El software codifica reglas. La inteligencia artificial produce borradores, clasificaciones o acciones. De ahí surge una inferencia seductora: si un job contiene tareas y las tareas pueden repartirse, el job puede descomponerse hasta desaparecer.

La economía laboral ofrece una lectura más cuidadosa. David Autor ha insistido en que las tecnologías sustituyen algunas tareas y complementan otras; las ocupaciones combinan ambas.¹ Daron Acemoglu y Pascual Restrepo muestran que la automatización desplaza tareas, pero la creación y reinstalación de tareas también importan.² Exposición no equivale a sustitución. Una ocupación no es una bolsa indiferenciada de unidades independientes.

La pregunta no es si podemos desagregar. Es **qué valor mantenía unido el bundle y cómo lo reconstruiremos**.

Job, task, workflow, capability, outcome

Cinco niveles suelen confundirse:

- un **job** agrupa responsabilidades, tiempo, relación, derechos y trayectoria;
- una **task** es una unidad de actividad;
- un **workflow** conecta tareas, decisiones, excepciones y handoffs;
- una **capability** permite producir de forma repetible bajo condiciones;
- un **outcome** es el resultado para un usuario o sistema.

Automatizar una task no significa automatizar el workflow.

Externalizar un workflow no significa transferir accountability. Eliminar una parte del job puede cambiar el aprendizaje que producía el resto.

En atención al cliente, por ejemplo, un sistema puede sugerir respuestas o recuperar conocimiento. El estudio de Brynjolfsson, Li y Raymond sobre 5.179 agentes encontró un aumento medio de productividad, medido como issues resueltos por hora, con mayor beneficio para agentes noveles y menos experimentados.³ El resultado muestra complementariedad en un contexto concreto. No demuestra que el puesto completo sea reemplazable. Alguien sigue gestionando casos, excepciones, empatía, escalado y responsabilidad.

Por qué ahora se puede separar más

La separabilidad aumenta cuando cambian cuatro fricciones.

Representación. Una tarea puede especificarse digitalmente mediante inputs y outputs.

Descubrimiento. Es más fácil encontrar una fuente especializada.

Transferencia. Datos y artefactos viajan sin copresencia.

Evaluación. El output puede compararse con un estándar o una prueba.

Estas condiciones explican por qué un reto de NASA puede aislar un problema o por qué una competición de datos puede recibir soluciones externas. También explican microservicios, traducción, pruebas o análisis puntuales. Pero una frontera técnica no elimina las dependencias humanas. La pieza separada debe volver a entrar en un sistema.

El Separability Test

Antes de extraer una tarea de un job, aplique siete pruebas.

1. MODULARIDAD

¿Los inputs, outputs y reglas de interfaz son suficientemente estables? ¿La pieza puede evolucionar sin cambiar continuamente las demás? Modular no significa simple; significa que las dependencias se gestionan mediante design rules.⁴

2. INTERDEPENDENCIA

¿Cuántos bucles cruzan la frontera? Si el executor necesita preguntar a cinco personas cada día, la tarea quizá solo parecía separable en el diagrama.

3. EVALUABILIDAD

¿Podemos reconocer un buen resultado antes de que cause daño? Un output fácil de medir admite más separación que una decisión cuyo error emerge meses después.

4. CONTEXTO

¿Qué conocimiento tácito, historia o relación necesita? ¿Puede transferirse sin consumir más trabajo que la ejecución?

5. ACCOUNTABILITY

¿Quién posee el outcome? Un sistema fragmentado puede asignar cada tarea y dejar la excepción sin dueño.

6. APRENDIZAJE

¿La tarea funciona como entrenamiento para capacidades futuras? Si automatizamos todo el trabajo junior, ¿cómo se forma la persona que

revisará casos complejos dentro de cinco años?

7. RELACIÓN Y PROTECCIÓN

¿Qué disponibilidad, identidad, carrera o protección estaba incluida en el job? Separar actividad económica puede trasladar riesgo o eliminar voz.

El resultado no es un sí/no permanente. Produce tres decisiones: **separar**, **mantener unido** o **rebundlear** en una nueva unidad coherente.

El impuesto de coordinación

Cada frontera nueva crea una interfaz. En el papel, un job se convierte en cuatro especialistas más eficientes. En la práctica, alguien define tareas, distribuye contexto, sincroniza decisiones, revisa outputs y resuelve lo que cae entre cajas.

El impuesto de coordinación tiene componentes observables:

- handoffs y espera;
- preguntas y aclaraciones;
- duplicación de contexto;
- revisión y retrabajo;
- discrepancias de herramientas o estándares;
- pérdida de memoria;
- conflictos de prioridad;
- escalados sin owner.

La separabilidad económica existe cuando el valor de especialización, flexibilidad o automatización supera ese impuesto y

no destruye aprendizaje o accountability esencial. La tarifa más baja no responde la pregunta.

Equipos temporales que sí funcionan

La investigación de Beth Bechky sobre crews cinematográficos muestra que organizaciones temporales pueden coordinarse sin una historia compartida larga.⁵ No funcionan porque todo sea líquido. Funcionan porque roles profesionales, reputación, rutinas y expectativas ya rebundlean parte de la coordinación. Gaffer, grip y otras funciones no llegan como individuos sin estructura; llegan dentro de una institución ocupacional.

El caso corrige una fantasía frecuente. Un equipo montado para una oportunidad no empieza desde cero. Se apoya en estándares, lenguaje, carreras y redes previas. Si una empresa quiere formar equipos dinámicos, debe construir o acceder a ese sustrato. La temporalidad visible descansa sobre permanencias invisibles.

IA y la frontera irregular

La IA generativa hace especialmente importante separar tarea y workflow. Un sistema puede redactar con rapidez y fallar en una premisa crítica. Puede mejorar output dentro de una frontera de capacidades y empeorarlo justo fuera. El experimento con consultores de BCG y GPT-4 mostró esa frontera irregular: desempeño superior en tareas dentro de las capacidades del modelo y deterioro en una tarea fuera de ellas.⁶

Por eso “automatizable” no es una propiedad de la tarea en abstracto. Depende de contexto, versión, herramientas, criterios y supervisión. Una tarea puede ser automatizable cuando el input está limpio, revisable y de bajo daño, y no serlo cuando contiene

excepciones o consecuencias severas.

La unidad de análisis debe incluir preparación y revisión. Si un modelo ahorra veinte minutos de redacción y añade treinta de verificación, no hemos ganado capacidad. Si permite que una persona novel resuelva más casos con igual calidad, sí puede haber un cambio. Si elimina las tareas con las que esa persona aprendía, debemos observar el efecto a más largo plazo.

Rebundling

Desagregar sin rebundlear produce componentes sin sistema.

Rebundling significa diseñar la nueva unidad de trabajo después de mover o automatizar piezas.

Debe responder:

1. ¿quién recibe el outcome completo?;
2. ¿quién gestiona excepciones?;
3. ¿cómo viaja el contexto?;
4. ¿quién revisa y con qué señal?;
5. ¿dónde se acumula memoria?;
6. ¿cómo se aprende y desarrolla expertise?;
7. ¿qué derechos, carga y riesgo cambian?;
8. ¿qué fallback existe?

Un rebundle humano-IA puede asignar al sistema recuperación y primer borrador, a la persona decisión y revisión, y a un owner la responsabilidad end-to-end. Un rebundle entre firmas puede mantener arquitectura y QA dentro, ejecución especializada fuera y un decision log compartido. No hay una forma única; hay una

obligación de cerrar el circuito.

El horizonte largo

Una prueba de tres meses puede mostrar coste y velocidad. No muestra necesariamente skill pipeline, dependencia o cultura. Después de 12–24 meses habría que medir:

- calidad y errores severos;
- retrabajo y escalado;
- time-to-productivity de personas nuevas;
- mentoring y adquisición de criterio;
- concentración de conocimiento;
- capacidad de recovery;
- experiencia y protección de quienes contribuyen.

La desagregación puede ser eficiente hoy y erosionar la capacidad de mañana. También puede liberar a personas de tareas rutinarias y mejorar aprendizaje si el nuevo bundle lo diseña. La dirección del efecto no se presume.

La objeción conservadora

Un test de separabilidad podría declarar inseparable todo trabajo complejo y proteger puestos heredados. El conocimiento tácito es fácil de invocar y difícil de medir. La respuesta es convertir cada barrera en condición de experimento: mejorar documentación, definir un umbral, pilotar una interfaz, observar handoffs y mantener fallback.

El test no debe congelar el job. Debe impedir que lo rompamos sin

saber qué mantenía unido. Si una dependencia puede reducirse, se reduce. Si el nuevo bundle funciona, la frontera cambia. Si el impuesto de coordinación supera el valor, se revierte.

Un puesto es una tecnología institucional. Como toda tecnología, puede estar sobredimensionado, mal diseñado o ser exactamente la solución eficiente. El objetivo no es conservarlo ni destruirlo. Es comprenderlo.

Al terminar esta parte ya podemos definir outcome, escoger modalidad y decidir la unidad. Pero una empresa no opera una capacidad aislada. Mantiene muchas, conectadas y expuestas a shocks comunes. **La siguiente tarea es ver la workforce como un portfolio, sin reducir a las personas a activos financieros.**

La organización vive en las excepciones

Los diagramas de procesos describen el camino normal. Los puestos suelen absorber lo que ocurre fuera de él. Una persona recuerda que un cliente tiene una condición especial, detecta que dos datos no encajan, negocia una prioridad, ayuda a alguien nuevo y decide cuándo una regla no debe aplicarse. Ese trabajo de excepción es difícil de inventariar porque se activa de manera irregular y se apoya en contexto.

Al desagregar un job, las tareas normales encuentran dueño con facilidad. Las excepciones quedan en las interfaces. Un proveedor entrega conforme al scope; una herramienta procesa la regla; un especialista completa su fragmento. Cuando aparece un caso no previsto, cada componente puede haber cumplido y el outcome completo fallar. La pregunta decisiva del *Separability Test* es entonces: ¿quién posee lo que todavía no sabemos describir?

La respuesta suele requerir un rol integrador con criterio y autoridad. No tiene por qué ejecutar todas las tareas. Debe observar el workflow, clasificar excepciones, modificar reglas y decidir cuándo cambiar la configuración. Si ese papel recae siempre en el founder o en un manager informal, la desagregación ha desplazado el puesto en lugar de eliminarlo.

El registro de excepciones es una fuente de diseño. Muestra qué tareas eran realmente separables, qué interfaces necesitan cambiar y qué conocimiento debe rebundlearse. Una empresa que solo mide throughput puede creer que fragmentó con éxito hasta que la variedad del mundo real supera sus reglas.

La multiplicación de interfaces

Dividir un bundle en n componentes no crea solo n piezas. Puede crear muchas relaciones entre ellas. Si cada especialista necesita coordinarse con todos los demás, el número potencial de conexiones crece mucho más deprisa que el número de ejecutores. En la práctica no todas las conexiones son necesarias, pero la intuición importa: la especialización ahorra ejecución y puede consumir integración.

Hay tres formas de contener ese crecimiento. La primera es una arquitectura modular con inputs, outputs y estándares estables. La segunda es un integrador que centraliza dependencias sin convertirse en cuello de botella. La tercera es un protocolo compartido —ritmos, herramientas, taxonomías y decision rights— que permite coordinar sin una conversación nueva para cada caso.

La mala respuesta es añadir reuniones. Una reunión puede resolver un episodio y esconder una interfaz defectuosa. Si la misma aclaración se repite, el sistema debe convertirla en una regla, una

decisión o una modificación del bundle. El objetivo no es reducir toda comunicación: el conocimiento tácito necesita interacción. Es distinguir conversación que crea aprendizaje de coordinación que paga una deuda recurrente.

Cinco patrones de rebundling

El rebundling no siempre devuelve el trabajo a un puesto tradicional. Puede adoptar varias formas.

Núcleo y satélites. Un owner estable conserva outcome y excepciones; fuentes especializadas ejecutan módulos. Funciona cuando el núcleo tiene criterio suficiente y las interfaces son observables.

Célula de capacidad. Un pequeño equipo combina funciones y tecnología alrededor de un outcome end-to-end. Reduce handoffs y puede conservar aprendizaje, a costa de duplicar algunas habilidades entre células.

Servicio con producto interno. Una capacidad se organiza como plataforma o servicio interno con catálogo, niveles y roadmap. Mejora acceso dentro de la empresa, pero puede crear distancia respecto al contexto local.

Partner comprometido más assurance interno. La ejecución reside fuera; una capacidad estable específica, revisa y aprende. Es útil cuando el mercado posee expertise superior y la institución debe conservar juicio.

Human-AI workflow. Tecnología ejecuta pasos delimitados; personas preparan, deciden, revisan y resuelven excepciones. Su valor depende de que la supervisión tenga señales y autoridad, no de insertar un humano nominal.

Cada patrón responde de manera distinta a volumen, aprendizaje, regulación y recuperación. El test no elige por nombre. Comprueba dónde quedan contexto, decisión y memoria.

La cantera de expertise

Los puestos no solo producen hoy. También forman a quienes producirán mañana. El trabajo junior puede parecer el candidato perfecto para automatizar o externalizar: es repetitivo, visible y caro de supervisar. Sin embargo, parte de ese trabajo construye patrones mentales, exposición a casos y comprensión del sistema. Si desaparece, la organización puede disfrutar de eficiencia inmediata y descubrir después que carece de personas capaces de revisar lo complejo.

El argumento no protege toda tarea de entrada. Muchas son tediosas y enseñan poco. Obliga a identificar qué experiencia produce criterio. Si una herramienta realiza el primer borrador, el diseño puede reasignar a la persona a comparación, diagnóstico y feedback. Si un proveedor ejecuta, puede incluir shadowing, rotación o revisión interna. Si el aprendizaje no puede reconstruirse, la capacidad necesita mayor proximidad.

Una métrica útil no es cuántas tareas junior quedan, sino cuánto tarda una persona en alcanzar autonomía sobre excepciones y qué evidencia de aprendizaje recibe. La automatización que reduce ese tiempo mejora el sistema. La que oculta el proceso y deja a la persona aprobando sin comprenderlo puede degradarlo.

Derechos y sentido del trabajo

Un job agrupa ingresos, identidad, relaciones, feedback, posibilidad de desarrollo y protección, además de tareas. Al separar actividad

económica, esas funciones no se redistribuyen automáticamente. La empresa puede ganar especialización y dejar a quienes contribuyen con fragmentos sin visibilidad del outcome ni oportunidades de aprendizaje. También puede ocurrir lo contrario: una persona obtiene más autonomía, variedad y control sobre su tiempo.

La evaluación debe observar quién puede planificar, decir no, comprender el uso de su trabajo y disputar una valoración. Estas dimensiones afectan a calidad y sostenibilidad, no son un apéndice ético. Una red de especialistas que recibe buen contexto y feedback puede producir mejor que una cadena de microtarefas opaca. Una arquitectura que intercambia personas como componentes reduce la inversión relacional de la que después depende.

Rebundlear derechos no significa recrear empleo en toda colaboración. Significa diseñar expectativas y mecanismos proporcionados a la duración, dependencia y consecuencias de la relación.

Separabilidad cambia con la tecnología y con la experiencia

Una actividad que hoy requiere mucha interacción puede volverse modular después de documentar, estandarizar o mejorar herramientas. Otra que parecía separable puede revelar excepciones y volver a integrarse. El test debe repetirse, no convertirse en un veredicto permanente.

La experiencia permite observar tres señales. **Interface load:** cuánto trabajo se dedica a cruzar la frontera. **Exception rate:** cuántos casos necesitan salir del camino normal y con qué daño. **Learning location:** dónde se acumula el conocimiento que mejora el siguiente ciclo. Si la primera y la segunda crecen o la tercera queda fuera de

control, el bundle necesita cambiar.

Esta visión evita discutir sobre ocupaciones enteras. La unidad es un sistema de tareas, decisiones y aprendizaje que evoluciona. Algunas piezas se automatizan, otras se especializan y otras se vuelven más valiosas porque integran. El futuro del job no es una resta mecánica. Es una sucesión de nuevos bundles.

El criterio final

La desagregación está justificada cuando el nuevo sistema produce mejor el outcome, conserva o mejora aprendizaje y distribuye derechos y riesgo de forma defendible. No basta con que cada fragmento sea más barato. Tampoco basta con que la tecnología pueda ejecutar una demo.

El test más honesto es observar el trabajo que aparece después de separar. ¿Cuánto contexto debe reintroducirse? ¿Quién resuelve excepciones? ¿Qué nuevas revisiones nacen? ¿Quién aprende? ¿Qué ocurre al fallar? Ese trabajo residual revela el bundle real. Si se hace visible, la empresa puede diseñarlo. Si permanece escondido, alguien lo absorberá hasta convertirse en cuello de botella.

Un job no es una lista de tareas. Si separa la ejecución, rebundlee contexto, excepciones, aprendizaje, derechos y accountability.

NOTAS DEL CAPÍTULO

1. David Autor, “**Why Are There Still So Many Jobs?**”, *Journal of Economic Perspectives*, 2015.
2. Daron Acemoglu y Pascual Restrepo, “**Automation and New Tasks**”, *Journal of Economic Perspectives*, 2019.
3. Erik Brynjolfsson, Danielle Li y Lindsey Raymond, “**Generative AI at Work**”, NBER Working Paper 31161. El estudio mide un deployment específico de customer support.
4. Carliss Baldwin y Kim Clark, *Design Rules, Volume 1*, MIT Press.
5. Beth Bechky, “**Gaffers, Gofers, and Grips**”, *Organization Science*.
6. Fabrizio Dell’Acqua et al., “**Navigating the Jagged Technological Frontier**”, 2023.

HERRAMIENTA — CAPABILITY CHOICE PACK

Complete las tres piezas sobre la misma capacidad.

A. BBBA DECISION RECORD

Outcome · propiedades · alternativas · combinación · coste total · riesgo bilateral · opción preservada · trigger · fecha.

B. FRACTIONAL FIT TEST

Volumen/cadencia · modularidad · contexto · disponibilidad · autoridad · conflicto · regulación · aprendizaje · owner end-to-end.

C. SEPARABILITY TEST

Mapa de tareas y decisiones · interfaces · excepciones · evaluabilidad · contexto · aprendizaje · derechos · nuevo bundle.

Si las herramientas recomiendan cosas distintas, no promedie. La contradicción suele señalar una unidad mal definida, un owner ausente o una dependencia que todavía no se ha hecho visible.

La modalidad bajo incertidumbre

Una empresa entra en un mercado nuevo. Necesita comprender regulación, adaptar producto, construir demanda y operar soporte. Puede contratar un equipo, desarrollar internamente a quienes ya tiene, utilizar especialistas, comprar servicios o automatizar pasos. La decisión real no es una matriz con cuatro columnas. Es una secuencia bajo incertidumbre.

Al principio desconoce volumen, excepciones y ritmo. Un puesto completo puede ser prematuro; un encargo spot puede no acumular contexto. Un partner committed puede comprar aprendizaje y opción. La automatización puede ayudar a explorar y también codificar supuestos débiles. La arquitectura correcta hoy puede ser incorrecta seis meses después.

El teorema de contingencia exige registrar qué no se sabe. Demanda, especificidad, daño, market depth, regulación y necesidad de aprendizaje forman una hipótesis. BBBA convierte esa hipótesis en combinación y trigger.

Diseñar el primer experimento

El primer experimento debería ser reversible y suficientemente real para revelar el workflow. Una demo sin usuarios no muestra excepciones; una externalización completa puede comprometer demasiado antes de aprender. La empresa puede conservar decision rights, utilizar expertise externo para explorar, asignar ejecución limitada y medir integración.

El outcome no es “lanzar el proyecto”. Es obtener evidencia sobre capacidad: qué tareas son separables, qué contexto domina, qué frecuencia emerge, qué daño importa y qué conocimiento merece control estable. El experimento produce un servicio y un mapa.

La elección de modalidad compra información. *Borrow* puede revelar el mercado y acelerar. *Build* puede revelar complejidad y formar criterio. *Buy* puede crear continuidad cuando el coste de espera es alto. *Automate* puede revelar variación y excepciones a gran escala. Compararlas solo por coste directo desperdicia su valor epistemológico.

Cuando aparece una necesidad fractional

Suponga que la regulación requiere criterio senior algunas horas por semana. La unidad parece fractional. El test pregunta si la demanda es previsible, si existe un owner interno y si una incidencia fuera de agenda necesita respuesta. Puede resultar un fit: advisor recurrente, sistema interno y escalado definido. Puede resultar una ficción: la empresa espera que la misma persona sea responsable en todo momento sin pagar reserva ni conceder autoridad.

La diferencia no es el número de horas. Es el objeto. Consejo, ejecución, construcción de sistema y accountability requieren relaciones distintas. El acuerdo debe decir cuál compra y cómo evolucionará.

La incertidumbre puede justificar lo fractional como fase, no como destino. Tras observar volumen y decisiones, la empresa puede contratar, profundizar la relación o reducirla. Un trigger evita que la provisionalidad se vuelva dependencia.

Cuando el job se rompe

Al adaptar producto, la organización descubre que un puesto combina investigación, decisión de producto, configuración técnica, comunicación y soporte. Distribuir cada tarea a la fuente más barata parece eficiente. Después aparecen handoffs y nadie posee la experiencia del cliente completa.

El Separability Test obliga a mapear excepciones y aprendizaje. Investigación puede abrirse a especialistas; decisión de producto puede permanecer en el Solid Minimum; configuración puede ejecutarse con partner; comunicación puede usar una red; soporte puede combinar automatización y personas. El nuevo bundle necesita un owner y una memoria que crucen todas las piezas.

El diseño se juzga por outcome end-to-end, no por eficiencia local. Si cada especialista cumple y el lanzamiento falla, la arquitectura es responsable.

El coste del cambio de modalidad

Reconfigurar tiene coste: contratación, salida, transferencia, permisos, herramientas, moral y relaciones. Una matriz estática omite esta histéresis. La empresa puede quedar en una modalidad subóptima porque salir cuesta más que tolerarla, o cambiar demasiado pronto y destruir el aprendizaje que empezaba a acumularse.

El Decision Record debe incluir *switching cost* y tiempo de transición. También los derechos que preservan opción: exportación, documentación, estándares, reserva y propiedad de artefactos. Una opción no elimina fricción; evita que sea desconocida.

La reversibilidad tiene un límite humano. Terminar una relación

afecta ingresos, confianza y reputación. El trigger debe ir acompañado de aviso y proceso, no utilizarse como algoritmo indiferente.

La prueba de la segunda mejor opción

Después de seis meses, la empresa compara resultado con el contrafactual que registró. Si eligió partner por velocidad, observa *time-to-capability*, management load y conocimiento. Si eligió empleo por continuidad, observa utilización, excepción y aprendizaje. Si automatizó por consistencia, observa calidad, review y severe errors.

No sabrá con certeza qué habría ocurrido. Pero puede descubrir si la razón original se cumplió. Esta disciplina reduce el *hindsight bias*: declarar que una decisión era correcta porque el resultado fue aceptable.

El mejor outcome no siempre corresponde a la modalidad más barata. Puede comprar opción estratégica, control o aprendizaje. La comparación debe hacer visible ese valor y también el coste de rigidez.

El veredicto de la Parte II

Una empresa Beyond Payroll no busca externalizar lo no core ni fragmentar todos los jobs. Busca escoger una unidad y una modalidad que correspondan a propiedades observables, mantener un owner y declarar cuándo cambiará.

La tesis fuerte de esta parte es negativa: no existe una regla universal. Su utilidad positiva está en el proceso. Outcome antes que solución; separabilidad antes que sourcing; mínimo sólido antes que

apertura; coste total y bilateral antes que ahorro; trigger antes que inercia.

La incertidumbre no desaparece. Se convierte en una hipótesis gestionable.

Cinco reversos de la tesis

Una idea se entiende mejor cuando obliga a actuar en la dirección contraria a su slogan. Si *Beyond Payroll* solo recomendara acceso, sus frameworks serían una racionalización. Los siguientes escenarios analíticos muestran cinco momentos en los que la decisión correcta puede ser construir, contratar, cerrar, simplificar o dejar de automatizar.

1. El proveedor sabe más que quien responde

Una empresa utiliza una firma especializada para una capacidad regulada. La ejecución es buena, los plazos se cumplen y el contrato asigna responsabilidades. Durante una revisión, el accountable executive descubre que no puede explicar los supuestos ni evaluar una recomendación sin ayuda de la misma firma.

La respuesta no tiene que ser terminar. Primero separa ejecución y juicio. Construye una pequeña capacidad interna, añade assurance independiente y exige artefactos que permitan comprender decisiones. Puede mantener al proveedor como fuente superior de expertise. Lo que internaliza es la capacidad de formular, aceptar y detener.

El reverso de acceso es **absorptive capacity**. Cuanto mejor sea el mercado, más fácil resulta comprar respuestas y más importante conservar criterio para distinguirlas.

2. La fracción se convirtió en función

Un profesional fractional comenzó con una cadencia limitada durante una fase incierta. Un año después participa a diario, posee decisiones, gestiona al equipo y responde fuera de horario. La demanda dejó de ser parcial; la relación conserva una etiqueta antigua.

El trigger obliga a revisar volumen, autoridad y dependencia. Puede conducir a empleo, a una función interna con advisor externo o a un managed service con equipo y cobertura. Mantener la fracción porque “funcionó” confunde éxito de entrada con fit permanente.

El reverso de *Fractional Everything* es reconocer cuándo la institución necesita continuidad y obligaciones recíprocas más completas.

3. El portfolio tiene demasiadas piezas

Una organización diversifica especialistas y proveedores. Cada decisión individual parece razonable. El founder pasa la mitad de la semana traduciendo, resolviendo conflictos y reconstruyendo contexto. Los costes directos son variables; la capacidad de decisión es fija.

La respuesta puede ser rebundlear. Reduce el número de fuentes, elige un partner committed para un conjunto coherente o incorpora un integrador estable. Renuncia a algo de variedad y gana una interfaz. El Total Access Cost cae aunque una tarifa suba.

El reverso de portfolio es **simplicidad deliberada**. La optionality que nadie puede gobernar no es opción efectiva.

4. La automatización eliminó la cantera

Un workflow automatiza tareas rutinarias de entrada. El throughput mejora. Dos años después, pocas personas comprenden el proceso end-to-end y los reviewers senior no tienen reemplazo. Las tareas eliminadas incluían exposición a patrones y excepciones.

La empresa no necesita volver al pasado. Rediseña aprendizaje: simulaciones, revisión explicada, rotaciones, casos adversos y participación gradual en decisiones. Puede reservar parte del trabajo para formar criterio o utilizar la automatización como tutor y no solo como sustituto.

El reverso de Automate es invertir deliberadamente en skill formation. La eficiencia presente no debe consumir la capacidad de supervisar el futuro.

5. La empresa abierta necesita cerrar

Una herramienta externa gana alcance mediante excepciones de permisos. Sus outputs son útiles y la dependencia crece. Después cambia el riesgo, aparece una regulación o se detecta un incidente. Cerrar acceso parece costoso porque datos, workflows y usuarios ya dependen.

La organización activa el exit design: exporta, revoca, opera un modo degradado y reconstruye. Puede sustituir, internalizar o reducir la capacidad. El cierre causa coste, pero preserva accountability.

El reverso de permeabilidad es la capacidad de decir no. Una frontera que solo se expande no está gobernada.

Lo que enseñan los reversos

Las cinco decisiones reducen acceso visible y pueden aumentar

capacidad efectiva. Construir criterio interno hace gobernable un proveedor. Convertir una relación en empleo alinea la función. Rebündlear reduce interfaces. Formar personas sostiene assurance. Cerrar una herramienta recupera control.

Esto corrige la interpretación lineal de la serie. Employment → Access → Orchestration no describe una evolución que abandona el término anterior. Describe nuevas opciones y el trabajo institucional que las limita. La organización puede moverse en ambas direcciones.

Una regla de honestidad

Todo framework debería poder recomendar su contrario aparente. BBBA debe poder recomendar *Buy*. Fractional Fit debe poder concluir que la fracción no encaja. Separability Test debe poder mantener el job unido. Capability Portfolio debe poder reducir variedad. Permeability Review debe poder cerrar.

Si la herramienta produce siempre apertura, no es diagnóstico. Es marketing.

La regla final es conservar un **reversion record**: qué se eligió, qué supuesto falló, qué coste apareció, quién absorbió la transición y qué cambia en el siguiente diseño. Sin ese aprendizaje, la empresa alterna centralización y externalización como modas. Con él, la frontera se vuelve una hipótesis acumulativa.

El valor estratégico de poder volver

Reversibilidad no significa ausencia de compromiso. Algunas capacidades requieren inversiones que solo producen valor con estabilidad. Significa comprender el coste de salida, preservar

derechos y detectar a tiempo cuándo el arreglo dejó de encajar.

Una empresa que puede volver es capaz de experimentar con mayor responsabilidad. No promete que cada opción sea barata. Evita que la velocidad de entrada cree una dependencia cuyo precio solo aparece al final.

PARTE III

LA FUERZA DE TRABAJO HÍBRIDA

Las decisiones individuales pueden parecer razonables y producir un sistema frágil. Esta parte cambia de escala: composición, concentración, arquitectura y mínimo sólido.

Capítulo 7

WORKFORCE AS A PORTFOLIO



Composición, opción y riesgo

Una empresa enumera cinco modalidades: empleados, freelancers, una agencia, un proveedor tecnológico y automatización. Parece diversificada. Después dibuja dependencias y descubre que la agencia, los freelancers y la automatización usan el mismo proveedor de cloud; que el founder conserva todo el contexto; y que cuatro piezas fallarían durante el mismo incidente. Cinco etiquetas,

una exposición.

El ejemplo es un stress test, no un caso disfrazado. Su función es mostrar por qué contar modalidades no basta. La diversidad contractual puede crear opción; también puede esconder concentración upstream. Un **Capability Portfolio** cambia la fila de la tabla. En lugar de listar personas o proveedores, lista capacidades y pregunta cómo se componen, de qué dependen y quién absorbe la variación.

La palabra portfolio merece cautela. Las personas no son activos financieros intercambiables. El objeto del portfolio es la capacidad. Las relaciones que la producen conservan derechos, identidad y condiciones específicas. Si el framework borra esas diferencias, habrá empeorado la gestión.

Tres denominadores

Una vista por headcount excluye trabajo externo y automatización. Una vista por gasto puede hacer que software barato parezca irrelevante aunque sea crítico. Una vista por outcomes puede ocultar quién trabaja y quién asume riesgo. Por eso el portfolio necesita tres denominadores simultáneos.

HORAS HUMANAS

Incluye empleo, founders, independientes, agencias y trabajo atribuible en proveedores cuando pueda estimarse. No pretende exactitud total; revela dónde existe trabajo invisible o concentración de una sola persona.

GASTO TOTAL

Incluye compensación, servicios, herramientas, coordinación,

integración, revisión, rework y salida. Las categorías contables se conservan, pero se superponen para la decisión.

CAPACIDAD Y OUTCOME

Registra volumen, latencia, calidad, fallos, recovery y aprendizaje. Dos modalidades con coste similar pueden producir distinta resiliencia.

Ninguna vista domina. Cuando cuentan historias diferentes, ahí está la investigación. Un proveedor puede representar poco gasto y mucho riesgo. Un founder puede representar cero salario y casi todas las decisiones. Una automatización puede reducir horas de ejecución y aumentar horas de review.

Las filas correctas

Cada fila del Capability Portfolio contiene:

- outcome y capacidad;
- accountable owner;
- criticidad;
- composición controlled / committed / accessible / automated;
- volumen y compromiso;
- Total Access Cost;
- tiempo de ramp;
- aprendizaje y ubicación de memoria;
- concentración por proveedor, plataforma, modelo, país y skill;
- supervisión y fallback;
- riesgo bilateral;
- trigger de revisión.

El portfolio no exige mapear cada actividad. Comience por las diez capacidades cuya pérdida o degradación afectaría más a clientes, cumplimiento o ventaja.

Composición no es diversificación

Una mezcla puede reducir coste fijo y permitir ajuste. Katharine Abraham y Susan Taylor encontraron que volatilidad, costes y skill requirements se relacionaban con el uso de contractors por las firmas.¹ La intuición es razonable: cuando la demanda cambia, una modalidad variable puede ofrecer opción.

Pero opción no equivale a resiliencia. Una empresa con tres agencias puede depender de la misma reserva de talento. Dos proveedores logísticos pueden compartir un cuello de botella geográfico. Varios modelos de IA pueden ejecutarse sobre la misma

infraestructura o degradarse ante la misma clase de input.

La pregunta es la **covarianza**: ¿qué fuentes fallan juntas? Los portfolios financieros diversifican cuando los riesgos no se mueven de la misma forma. En capacidades, los shocks comunes incluyen:

- escasez de una skill;
- interrupción de cloud o plataforma;
- cambio regulatorio;
- subida de precio de compute o materiales;
- cierre de frontera o problema de pagos;
- pérdida del founder que integra;
- demanda simultánea entre clientes;
- fallo de una dependencia open source.

Una segunda fuente solo es fallback si puede operar bajo el escenario que inutiliza la primera.

Automattic y las fronteras de comunidad

Automattic y el ecosistema WordPress permiten ver categorías que un organigrama mezcla con facilidad. La compañía tiene empleados; WordPress también depende de contributors, extensiones, hosting, agencias y una comunidad más amplia. Que una comunidad produzca valor no la convierte en workforce controlada por una empresa. Tampoco elimina preguntas de governance, mantenimiento o apropiación.

El portfollio debe distinguir:

- trabajo contratado por la firma;
- contribución comunitaria autónoma;
- proveedores que sirven a múltiples actores;
- infraestructura compartida;
- capacidad que la empresa puede movilizar de forma fiable.

Contar a toda la comunidad como “talento accesible” inflaría la frontera. Ignorarla escondería una dependencia. La categoría correcta puede ser: **ecosistema relevante, capacidad no comprometida.**

Concentraciones que no aparecen

Las concentraciones más peligrosas suelen estar en otra capa que la relación formal.

Founder concentration. Varias personas ejecutan, una sola integra y decide.

Knowledge concentration. El proveedor es sustituible en contrato, pero nadie más comprende las excepciones.

Platform concentration. Muchas fuentes se descubren, pagan o evalúan en el mismo intermediario.

Model concentration. Distintas aplicaciones dependen de una familia de modelos o del mismo proveedor.

Geographic concentration. Una red global está agrupada en una jurisdicción o huso crítico.

Data concentration. Todas las capacidades necesitan un dataset cuyo owner, calidad o permiso es único.

El portfolio no elimina estas exposiciones. Hace posible discutir las antes del incidente.

Bilateral Performance

Una firma puede aumentar su elasticidad haciendo que el ingreso de otras personas sea más volátil. Puede pagar solo horas ejecutadas y trasladar disponibilidad, formación, equipo y espera. Desde su dashboard, el coste se ajusta a la demanda. Desde el otro lado, la variación se ha concentrado.

La OIT ha descrito cómo algunos modelos de plataforma trasladan capital y costes operativos hacia quienes trabajan, además de usar gestión algorítmica.² El mecanismo no se limita a plataformas y no condena todo contracting. Obliga a añadir una segunda performance:

- predictibilidad de trabajo e ingreso;
- tiempo no facturado;
- gastos asumidos;
- capacidad de rechazo y salida;
- concentración en un cliente;
- acceso a aprendizaje y protección;
- claridad y apelación en la evaluación.

Una arquitectura no es sostenible si su “flexibilidad” depende de una contraparte permanentemente sobreexpuesta. La empresa puede aceptar cierta asimetría por la naturaleza del mercado, pero debe verla y gobernarla.

El shock correlacionado

La fabricación de aviación volvió visibles dependencias que existían antes del shock. El informe de GAO sobre producción posterior a 2020 recoge escasez de workforce y materiales entre fabricantes y proveedores.³ No prueba que más internalización hubiera resuelto todo. Prueba que una red de suppliers puede carecer de capacidad adicional cuando todos enfrentan la misma restricción.

Para cada capacidad crítica, ejecute tres escenarios:

1. demanda propia se duplica;
2. demanda del mercado se duplica para todos;
3. la fuente principal desaparece treinta días.

La primera prueba mide ajuste. La segunda, prioridad real. La tercera, recovery. Una reserva que funciona solo cuando nadie más la necesita no es equivalente a committed capacity.

Optionality y slack

Un portfolio robusto puede mantener:

- acuerdos de reserva;
- second source verdaderamente independiente;
- documentación y portabilidad;
- capacidad interna mínima para evaluar;
- slack deliberado en cuellos de botella;
- derechos de exportación y salida;
- pruebas periódicas de fallback.

Estas opciones cuestan. Pagar por redundancia puede parecer ineficiente en periodos tranquilos. No pagarla convierte el ahorro en

una apuesta. La decisión correcta depende del daño y del tiempo de recuperación, no de una preferencia estética por lo “lean”.

No existe el mix ideal

El portfolio invita a pedir una cifra: ¿qué porcentaje debería ser empleado, externo o automatizado? Esa cifra sería falsa. Un hospital, una fábrica y una empresa de software enfrentan restricciones distintas. Dentro de una misma empresa, identidad, ciberseguridad, traducción y diseño no necesitan el mismo control.

El target no es una composición contractual. Es un conjunto de capacidades que cumple outcomes con coste total, aprendizaje, responsabilidad y riesgo aceptables.

La utilidad del portfolio se prueba si cambia una decisión: descubre una concentración, justifica una reserva, internaliza un mínimo, retira una automatización o mejora condiciones porque la variación estaba mal distribuida. Si solo produce un gráfico, es reporting adicional.

Ahora vemos qué cambia y de qué depende. Falta decidir qué **no** debería desaparecer durante el cambio. El siguiente capítulo abandona la cómoda etiqueta *core* y pregunta por el **Solid Minimum**.

Un portfolio tiene tesis, límites y rebalanceo

Llamar *portfolio* a una colección de modalidades no basta. Un portfolio real expresa una tesis sobre qué papel cumple cada fuente, cuánto riesgo acepta la empresa y qué señal obligará a cambiar la composición. Una relación estable puede aportar memoria; una red committed, capacidad reservada; el mercado, variedad; la automatización, consistencia y escala en pasos delimitados. Si dos

fuentes cumplen exactamente la misma función, la segunda puede ser redundancia. Si solo parecen distintas por su contrato, quizá no diversifique nada.

La ficha de cada capacidad necesita una **función de portfolio**. Puede ser núcleo de continuidad, reserva para picos, exploración, second source, laboratorio, assurance independiente o automatización de volumen. La función permite evaluar si la configuración hace lo que se esperaba. Un proveedor de reserva que nunca se prueba no es fallback; es una línea contractual. Una red exploratoria que termina sosteniendo operación crítica ha cambiado de papel y requiere otra gobernanza.

El portfolio también tiene límites de concentración. No todos deben expresarse como porcentajes rígidos. Puede haber un máximo de decisiones críticas en una sola persona, una exigencia de exportabilidad, un tiempo de recuperación aceptable o una prohibición de que ejecución y assurance residan en la misma fuente. Lo importante es que el límite exista antes del incidente.

Por último, necesita rebalanceo. Una revisión trimestral o semestral observa demanda, performance, aprendizaje, dependencia y condiciones bilaterales. No reemplaza la gestión cotidiana; corrige acumulaciones silenciosas. La organización puede profundizar una relación, construir capacidad interna, retirar una herramienta o aceptar deliberadamente una concentración porque el coste de mitigación supera el riesgo.

Cuatro vistas que no deben fusionarse

Un único dashboard invita a sumar dimensiones incompatibles. Es más útil conservar cuatro vistas relacionadas.

La **vista de producción** muestra outcome, volumen, calidad, latencia y recuperación. Responde si el sistema cumple. La **vista de recursos** reconstruye trabajo humano, gasto, capital, software e inputs. Responde qué sostiene el resultado. La **vista de dependencia** mapea concentración, sustitución, derechos, ubicación de memoria y shocks comunes. Responde cómo puede fallar. La **vista bilateral** muestra distribución de volatilidad, costes, autonomía, aprendizaje y protección. Responde quién absorbe el diseño.

Una capacidad puede aparecer verde en producción y roja en dependencia. Eso no es una inconsistencia que deba promediarse. Significa que funciona hoy y puede fallar de forma material. Otra puede ser robusta para la empresa y mala para la contraparte; el riesgo social, reputacional o regulatorio puede tardar en entrar en los indicadores operativos. Una tercera puede costar más y conservar la opción estratégica que justifica ese gasto.

Las cuatro vistas previenen dos simplificaciones. La primera es convertir cualquier resultado en una puntuación de madurez. La segunda es utilizar la complejidad como excusa para no decidir. Cada divergencia debe terminar en una acción o en una aceptación explícita del riesgo.

Dependencias de segundo orden

El mapa directo suele registrar proveedor, persona y herramienta. Las dependencias importantes pueden estar un nivel más lejos. Una agencia y varios freelancers utilizan el mismo sistema de diseño; distintas aplicaciones invocan el mismo modelo; varios proveedores de pagos dependen de una red; una comunidad abierta confía en pocos maintainers; equipos distribuidos necesitan una plataforma de identidad común.

El primer paso es preguntar a cada fuente: “¿de qué depende para cumplir nuestra promesa?”. No se necesita reconstruir toda la economía. Se necesita seguir la cadena hasta encontrar puntos cuyo fallo afectaría a varias capacidades a la vez. Esos puntos son **common-mode dependencies**.

La segunda orden también incluye dependencia de coordinación. Cinco proveedores aparentemente independientes pueden esperar decisiones de una misma persona interna. Si esa persona falta, las cinco fuentes se detienen. El portfolio diversifica ejecución y concentra integración. Esta exposición explica por qué organizaciones con muchas opciones externas pueden seguir siendo poco elásticas.

El tratamiento depende del daño. Puede bastar con distribuir conocimiento, documentar decisiones o nombrar un segundo integrador. Puede exigir proveedores sobre infraestructuras distintas, una capacidad manual de emergencia o un cambio de arquitectura. La duplicación nominal es la respuesta más cara y, a veces, la menos efectiva.

El precio de la opción

La optionality no es gratuita. Reservar disponibilidad, mantener un second source, conservar skills internos o diseñar portabilidad consume dinero y atención. Las organizaciones tienden a recortarlo porque el beneficio es contrafactual: se observa el coste cada mes y solo se observa el valor cuando ocurre un shock.

La decisión puede tratarse como una prima. ¿Qué evento cubre la opción? ¿Qué pérdida evita? ¿Con qué probabilidad aproximada y durante qué periodo? ¿Qué parte de la recuperación compra realmente? No hace falta fingir una precisión actuarial. Sí hace falta

distinguir una reserva útil de una tranquilidad simbólica.

Una opción puede además degradarse. El proveedor alternativo deja de conocer el sistema, la persona interna no practica, el proceso manual queda obsoleto o la exportación nunca se prueba. Por eso la prima incluye ejercicios. Un *fallback* no es un documento, sino una ruta que alguien ha recorrido con suficiente frecuencia.

La empresa puede decidir no pagar. Para una capacidad de bajo daño y mercado profundo, aceptar latencia puede ser racional. El error no es carecer de redundancia. Es denominar resiliente a una configuración que eligió eficiencia de estado estable.

Portfolio y estrategia de talento

La vista de capacidades cambia también el desarrollo interno. En lugar de planificar carreras solo por puestos, la empresa puede identificar qué conocimiento necesita distribuir, qué roles integradores son escasos y qué experiencias crean criterio. Una persona empleada puede participar en varias capacidades; un partner puede aportar una; un workflow humano-IA puede redistribuir tareas. El portfolio revela dónde conviene invertir en aprendizaje, no solo dónde falta volumen.

Esta perspectiva evita que “acceder” se convierta en sustituto de desarrollar. Un mercado externo profundo puede resolver picos y especialización, pero la empresa necesita personas capaces de especificar, evaluar y aprender. Si cada capacidad nueva llega terminada desde fuera, el portfolio crece en variedad y se vacía en juicio.

El objetivo no es formar internamente todas las skills. Es construir **absorptive capacity**: lenguaje, criterio y relaciones

suficientes para reconocer valor y convertirlo en práctica. Puede residir en un equipo pequeño, distribuido entre funciones o compartido con una red committed. Debe estar identificado.

Gobernanza del portfolio

HR, procurement, tecnología, finanzas y riesgo suelen observar cortes distintos. Nadie posee la configuración completa. La solución no requiere crear un “director de todo el talento” que centralice cada decisión. Requiere un protocolo común y ownership por capacidad.

El accountable owner mantiene la ficha de outcome, composición, dependencias y triggers. Las funciones especialistas aportan reglas: clasificación y desarrollo, contratación y proveedores, seguridad y datos, presupuesto y controles. Una revisión de portfolio se centra en las capacidades de mayor materialidad y en excepciones, no en aprobar cada encargo.

La gobernanza debe ser proporcional. Un diseñador spot para una pieza pública no necesita el mismo comité que un proveedor con acceso a datos sensibles. La materialidad puede evaluarse por daño, irreversibilidad, concentración, aprendizaje y responsabilidad. Así el portfolio evita convertirse en burocracia universal.

El resultado de una buena revisión es una decisión: eliminar una concentración, financiar una opción, mejorar una interfaz, trasladar memoria, cambiar condiciones o aceptar el riesgo. Si la reunión termina solo con colores actualizados, el portfolio ha perdido su función.

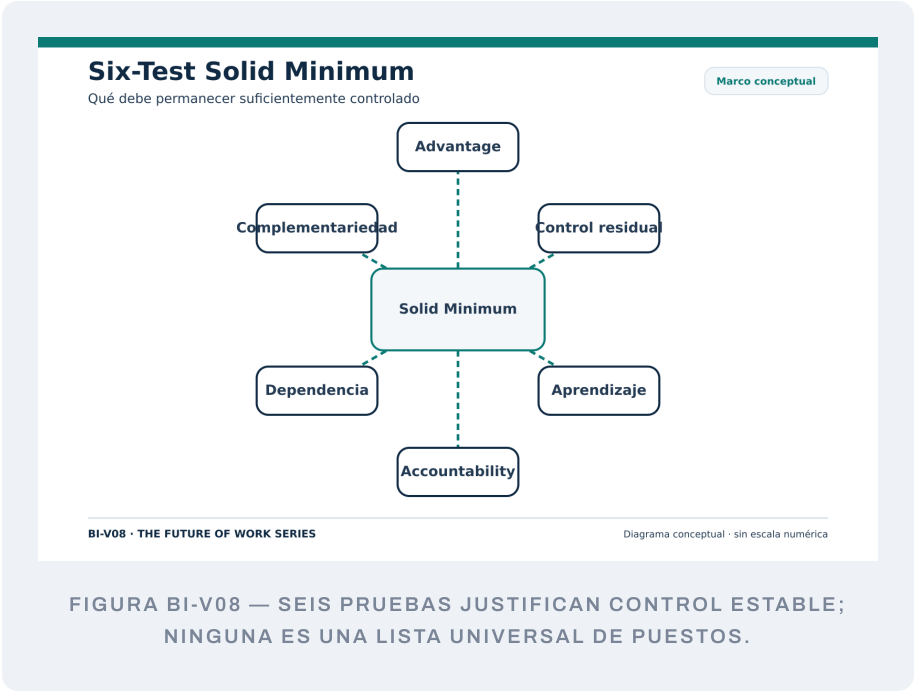
Una mezcla de contratos no es un portfolio robusto.
La diversificación existe cuando cambian las
dependencias y el sistema conserva fallback sin
ocultar quién absorbe el riesgo.

NOTAS DEL CAPÍTULO

1. Katharine G. Abraham y Susan K. Taylor, “Firms’ Use of Outside Contractors”, NBER Working Paper 4468.
2. OIT, *The Role of Digital Labour Platforms in Transforming the World of Work*, 2021.
3. U.S. GAO, *Commercial Aviation Manufacturing*, 2024.

Capítulo 8

WHAT MUST STAY CLOSE?



¿qué debe sobrevivir para que la empresa conserve ventaja, juicio, aprendizaje, continuidad y responsabilidad?

La fabricación de aviación hace visible la diferencia entre ejecución y responsabilidad. Un fabricante puede depender de una extensa red de suppliers; la calidad, certificación, integración y confianza del cliente no viajan fuera con cada contrato. Los problemas de capacidad y materiales documentados por GAO muestran que la red no elimina el cuello de botella ni la obligación del integrador.¹

No se deduce que todo deba internalizarse. Se deduce que cada frontera necesita un mínimo sólido.

Solid Minimum

El **Solid Minimum** es el conjunto mínimo de derechos, relaciones, conocimiento, activos, redundancias y rutinas que debe permanecer suficientemente estable para que una capacidad pueda cambiar de configuración sin perder lo esencial.

La palabra mínima importa. Sin ella, “control” se expandiría hasta incluirlo todo. La palabra sólida tampoco significa inmóvil. Una arquitectura puede actualizarse; lo que no puede hacer es desaparecer sin que la empresa pierda la capacidad de decidir y responder.

El **Solid Minimum Test** contiene seis pruebas.

1. Advantage

¿La capacidad contribuye a una diferencia que la empresa puede crear y apropiarse? Jay Barney argumentó que recursos valiosos, raros, difíciles de imitar y organizados pueden sostener ventaja.² La

lección para una economía de acceso es incómoda: si cualquier competidor accede a la misma fuente en condiciones similares, el acceso por sí solo no es ventaja.

La ventaja puede residir en cómo se combina la capacidad: datos propios, workflow, relación con clientes, velocidad de aprendizaje o criterio. La empresa no tiene que ejecutar cada pieza, pero necesita controlar el mecanismo que hace distinta la combinación.

Mínimo posible: arquitectura, decision right, dataset, relación o rutina de integración; no necesariamente toda la ejecución.

2. Residual control

Los contratos son incompletos. No anticipan cada estado del mundo. Grossman, Hart y Moore explican que los derechos residuales importan cuando aparece lo no previsto.³

Pregunte: ¿quién puede cambiar prioridad, detener el sistema, reasignar un activo o decidir ante una excepción? Si ese derecho está completamente fuera, la empresa puede haber externalizado más que ejecución.

Mínimo posible: owner con autoridad, derechos de step-in, acceso a artefactos, reglas de escalado y control de una dependencia crítica.

3. Cumulative learning

Algunas capacidades mejoran porque cada ciclo deja memoria. La firma, según la knowledge-based view, integra conocimiento especializado que ninguna persona posee por completo.⁴ Si cada contributor se marcha con contexto, la organización compra outputs y no acumula capacidad.

El mínimo no exige que todo conocimiento sea interno. Exige que la empresa pueda formular mejores preguntas, evaluar calidad y retener decisiones, excepciones y aprendizajes. Un partner externo puede formar parte de esa memoria mediante relación comprometida; el riesgo aparece si la memoria solo existe en su equipo y el contrato no garantiza transferencia.

Mínimo posible: knowledge owner, decision log, evaluación interna y ritual de transferencia.

4. System accountability

¿Qué daño, obligación o promesa permanece en la empresa aunque otra parte ejecute? Seguridad, privacidad, calidad, discriminación, cumplimiento y continuidad no se externalizan moralmente con una factura.

La accountability necesita alguien capaz de comprender el sistema, no solo firmar. Si el owner carece de acceso, criterio o autoridad para intervenir, la responsabilidad es nominal.

Mínimo posible: accountable executive, policy, assurance, audit trail, incident response y poder de bloqueo.

5. Bottleneck and dependence

Una capacidad puede ser común y convertirse en crítica por concentración. Bus factor, proveedor único, plataforma, modelo o geografía crean dependencia. La pregunta no es solo cuánto cuesta la fuente, sino cuánto cuesta perderla y cuánto tarda el recovery.

El mínimo puede ser redundancia, portabilidad o una pequeña capacidad interna capaz de operar durante la transición. Internalizar todo el proveedor rara vez es viable; reducir la irreversibilidad sí.

Mínimo posible: second source, exportabilidad, escrow, estándares, inventario de dependencias y pruebas de recuperación.

6. Complementarity

Datos, IP, marca, relaciones y procesos pueden producir más valor juntos que por separado. Una decisión que optimiza una pieza puede dañar la combinación. Holmström y Roberts subrayaron que complementariedades e incentivos complican reglas mecánicas sobre fronteras.⁵

Si una capacidad está coespecializada con otra, quizá el mínimo sea una interfaz profunda o una relación duradera. Separar contratos no separa la economía del sistema.

Mínimo posible: integración compartida, governance conjunta, estándares y decisiones coordinadas.

Apple: control selectivo, no ausencia de activos

Apple es utilizada con frecuencia como ejemplo de producción externalizada. La descripción resulta incompleta si sugiere una firma hueca. La compañía controla diseño, plataforma, software, marca, relaciones y decisiones centrales mientras trabaja con una extensa cadena de suministro.⁶ Su escala y poder contractual son excepcionales; otras empresas no pueden copiar la frontera simplemente imitando la etiqueta “asset-light”.

El caso ilustra control selectivo. La fabricación externa no elimina la necesidad de supplier standards, ingeniería, supervisión, capacidad de negociación e integración. También muestra una limitación: una firma puede controlar elementos estratégicos y seguir expuesta a concentración geográfica, reputación o capacidad

de proveedores.

La lección no es “externalice manufactura”. Es “identifique qué combinación debe seguir bajo control y qué capacidades hacen gobernable la red”.

Red Hat, Linux y producción abierta

Red Hat construyó negocio alrededor de software abierto producido por comunidades e instituciones diversas. El valor no procede de poseer cada contribución. Puede residir en integración, soporte, seguridad, certificación, confianza y una distribución empresarial gobernada.⁷

El caso complica dos clichés. Primero, lo abierto no significa ausencia de empresa. Segundo, el control no significa cerrar todo el conocimiento. Una institución puede permitir producción abierta y conservar un mínimo sólido alrededor de assurance, relación y responsabilidad.

La frontera correcta no coincide con el límite de propiedad intelectual. Coincide con los elementos que permiten convertir producción distribuida en una promesa fiable al cliente.

Formas de control

La discusión inside/outside empobrece las opciones. El control estable puede tomar distintas formas:

NECESIDAD	FORMA POSIBLE DE CONTROL
Decision right	owner interno, veto, step-in
Disponibilidad	empleo, retainer, reserva, capacidad dedicada
Apropiación	propiedad, licencia, secreto, datos, combinación
Calidad	estándar, certificación, audit right, sampling
Continuidad	redundancia, portabilidad, escrow, fallback
Aprendizaje	documentación, rotación, pairing, review interno
Relación	exclusividad limitada, tenure, governance conjunta

El objetivo es elegir la forma menos rígida que proteja el requisito. Si un derecho de exportación y una prueba de fallback resuelven la dependencia, quizá no haga falta internalizar. Si una capacidad de evaluación interna no basta ante daño severo, el control deberá aumentar.

Reversión como conocimiento

Las narrativas empresariales suelen presentar outsourcing e insourcing como direcciones ideológicas. En realidad, la reversión puede revelar aprendizaje. Una firma externaliza, descubre que la tarea contiene más conocimiento tácito o dependencia de lo esperado y cambia el arreglo. Otra internaliza, descubre que el mercado ofrece mejor especialización y abre una parte.

El Decision Record del capítulo 4 permite distinguir dos casos. Si

las propiedades cambiaron, la decisión anterior pudo ser correcta. Si estaban mal comprendidas, el error enseña qué test falló. En ambos, la reversión es parte del sistema, no una vergüenza.

La objeción: el mínimo que crece

Cada departamento puede usar las seis pruebas para declarar que su capacidad debe permanecer cerca. Seguridad invoca accountability; producto, aprendizaje; operaciones, bottleneck. El Solid Minimum acabaría siendo la empresa entera.

La disciplina está en nombrar el objeto mínimo, no la función completa. “Marketing debe estar dentro” es una conclusión perezosa. “La empresa debe conservar decision rights de posicionamiento, datos de cliente, criterio de marca y capacidad de evaluar partners” es falsable. Parte de la ejecución puede seguir en otras modalidades.

También hay que registrar el **coste de rigidez**:

- capital fijo;
- menor variedad externa;
- contratación lenta;
- riesgo de obsolescencia;
- autoridad concentrada;
- pérdida de opciones.

El mínimo se justifica cuando reduce un riesgo material más que la rigidez que crea.

La pregunta de supervivencia

Una manera simple de aplicar el test es imaginar la salida de la fuente actual mañana.

- ¿Qué conocimiento se va?
- ¿Qué decisión queda sin dueño?
- ¿Qué obligación sigue siendo nuestra?
- ¿Qué activo no podemos usar?
- ¿Qué cliente deja de confiar?
- ¿Cuánto tardamos en recuperar?

Lo que debe sobrevivir define el mínimo sólido. No siempre será una persona. Puede ser un protocolo, un derecho, una relación, un repositorio, una redundancia o una pequeña célula interna. Con frecuencia será una combinación.

El portfolio nos mostró composición y riesgo. El Solid Minimum decide qué debe permanecer. Ahora falta una arquitectura que conecte outcome, workflow, decision rights, modalidad, interfaz, assurance y memoria. Sin ella, las decisiones seguirán viviendo en documentos separados. **Esa es la nueva talent architecture.**

El mínimo sólido es específico del escenario

Preguntar qué debe permanecer “dentro” sigue siendo demasiado general. Una capacidad puede necesitar un mínimo para operar en condiciones normales y otro para sobrevivir a una crisis. En estado estable, un owner, una interfaz y assurance pueden bastar. Ante un incidente severo, la empresa quizá necesite acceso directo a datos, derecho de intervención, personas disponibles y una ruta alternativa.

El **Solid Minimum** debe declarar el escenario que protege. Puede

ser continuidad diaria, recuperación de treinta días, defensa de propiedad intelectual, cumplimiento en una jurisdicción o conservación de aprendizaje durante un cambio de proveedor. Sin escenario, toda función puede justificar máxima proximidad.

Esta especificidad permite combinar mínimos. Un *operating minimum* sostiene el workflow; un *recovery minimum* permite restablecerlo; un *strategic minimum* conserva la diferencia y el aprendizaje; un *accountability minimum* garantiza que la institución pueda decidir y responder. A veces coinciden. A veces requieren objetos distintos.

La capacidad de juzgar

La forma más fácil de vaciar una empresa no es externalizar ejecución. Es perder la capacidad de formular y evaluar. Una organización puede seguir tomando decisiones formalmente mientras depende de la contraparte para explicar qué opciones existen, qué calidad es aceptable y qué riesgo importa. El owner interno firma; el juicio reside fuera.

Conservar juicio no exige replicar el expertise completo del proveedor. Exige competencia suficiente para definir el outcome, cuestionar supuestos, interpretar evidencia, reconocer límites y detener. En una capacidad técnica compleja, esa competencia puede ser un arquitecto interno, un panel independiente o una segunda fuente de assurance. En una función creativa, puede ser criterio de marca y conocimiento de cliente. En regulación, una persona responsable con acceso a asesoramiento especializado.

La prueba es contrafactual: si la fuente recomienda ampliar su propio alcance, ¿puede la empresa evaluarlo sin pedirle que se evalúe a sí misma? Si un output parece correcto, ¿quién sabe qué

evidencia falta? Si hay desacuerdo entre proveedores, ¿existe un principio de decisión? Cuando las respuestas son negativas, el mínimo sólido necesita reforzar juicio antes que ejecución.

Control por diseño, contrato y capacidad

El control estable adopta tres familias. **Control por diseño** limita dependencias mediante modularidad, estándares, portabilidad y permisos. **Control contractual** asigna reserva, derechos de auditoría, propiedad de artefactos, step-in, salida y responsabilidad. **Control por capacidad** mantiene personas, relaciones y conocimiento capaces de operar o evaluar.

Las tres se complementan. Un contrato de exportación sirve poco si la arquitectura no puede utilizar los datos. Una interfaz modular sirve poco si nadie conoce el dominio para sustituir. Un equipo interno sirve poco si carece de derechos sobre la infraestructura. Diseñar el mínimo significa elegir la combinación menos rígida que funciona bajo el escenario.

Esta descomposición es especialmente útil para empresas pequeñas. No pueden internalizar cada función crítica. Sí pueden conservar cuentas maestras, documentación, rights de acceso, un responsable owner y una red probada. Pueden evitar que todas las credenciales pertenezcan a una agencia, que los datos residan en sistemas inexportables o que una sola persona externa sea la única capaz de explicar.

Apropiación y aprendizaje

Acceder a una capacidad común puede mejorar performance y no crear ventaja. Si todos los competidores contratan el mismo talento, usan el mismo software y consultan el mismo modelo, la diferencia

se desplaza hacia la combinación. Datos propios, relaciones, decisiones, velocidad de aprendizaje y confianza pueden seguir siendo difíciles de imitar.

El Solid Minimum debe proteger el **learning loop** que convierte acceso común en práctica singular. Eso incluye formular mejores problemas, registrar qué funcionó, modificar interfaces, integrar feedback de clientes y distribuir conocimiento. No necesita poseer cada idea. Necesita conservar la capacidad de recombinarlas y de aprender antes que otros.

La apropiación tampoco es absoluta. Una comunidad, partner o profesional conserva legítimamente su conocimiento y puede crear valor para múltiples actores. La empresa debe distinguir lo que necesita proteger —datos, secretos, decisiones, relación— de aquello cuya apertura produce ecosistema y aprendizaje. Cerrar demasiado puede destruir la fuente de innovación que pretendía controlar.

Sectores que cambian la respuesta

En software digital, modularidad y reversibilidad pueden permitir un mínimo relativamente pequeño alrededor de arquitectura, datos, seguridad y producto. En manufactura, activos físicos, calidad y supply chain añaden tiempos de recuperación largos. En salud o servicios profesionales regulados, licencias, deberes y daño potencial elevan el accountability minimum. En operaciones locales, presencia y conocimiento del contexto pueden dominar.

Incluso dentro del sector, la etapa importa. Una startup puede acceder a expertise para explorar y mantener founder judgment. Al crecer, volumen, regulación y repetición pueden justificar control más estable. Una empresa madura puede abrir una capacidad que ya estandarizó. Copiar la frontera de una compañía famosa sin copiar

su escala, poder contractual y sistemas conduce a una arquitectura distinta de la que se imagina.

La pregunta correcta es siempre local: ¿qué perderíamos, cuánto tardaríamos en recuperarlo y qué forma de control resuelve esa pérdida con menor rigidez?

El coste de un mínimo demasiado grande

El riesgo de fragilidad es visible después de externalizar; el riesgo de rigidez puede crecer en silencio. Internalizar por precaución inmoviliza capital, reduce variedad, alarga contratación y protege prácticas que quizá deberían cambiar. Un equipo puede defender su existencia mediante conocimiento que solo él controla, creando dependencia interna.

Cada objeto del mínimo debe tener un **coste de solidez** y un trigger de revisión. ¿Qué opción perdemos al mantenerlo? ¿Qué demanda mínima justifica el compromiso? ¿Qué mercado o tecnología podría volverlo innecesario? ¿Qué evidencia demostraría que el control no mejora outcome ni recovery?

Esta prueba preserva la palabra “minimum”. El objetivo no es maximizar lo interno, sino evitar una empresa hueca sin convertirla en una fortaleza cerrada.

El mínimo como promesa institucional

La empresa existe también porque alguien puede comprometerse a cumplir y responder. El mínimo sólido sostiene esa promesa cuando la ejecución se distribuye. Incluye quién puede hablar en nombre de la organización, quién tiene autoridad para detener, qué memoria permite explicar una decisión y qué recursos se movilizan para

reparar.

Desde esta perspectiva, el mínimo no es una lista de activos “core”. Es una arquitectura de responsabilidad. Puede contener empleo, propiedad, contratos y partners. Su prueba final no es dónde está cada recurso, sino si la institución conserva capacidad para cumplir su palabra en estados normales y adversos.

Lo que debe permanecer cerca es, en última instancia, aquello sin lo cual la empresa ya no puede distinguir, aprender, decidir o responder. Todo lo demás sigue abierto a comparación.

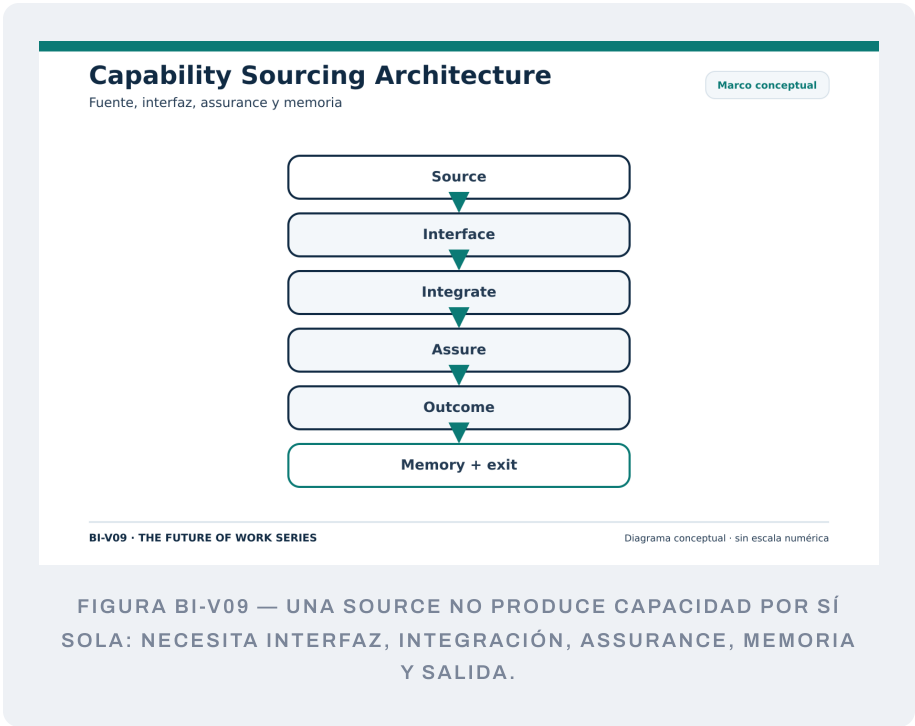
No pregunte qué es core. Pregunte qué debe sobrevivir a la salida de la fuente actual y cuál es la forma menos rígida de conservarlo.

NOTAS DEL CAPÍTULO

1. U.S. GAO, *Commercial Aviation Manufacturing*, 2024.
2. Jay Barney, “Firm Resources and Sustained Competitive Advantage”, *Journal of Management*, 1991.
3. Sanford J. Grossman y Oliver D. Hart, “The Costs and Benefits of Ownership”, 1986; Oliver Hart y John Moore, “Property Rights and the Nature of the Firm”, 1990.
4. Bruce Kogut y Udo Zander, “Knowledge of the Firm...”, 1992; Robert Grant, “Toward a Knowledge-Based Theory of the Firm”, 1996.
5. Bengt Holmström y John Roberts, “The Boundaries of the Firm Revisited”, 1998.
6. Apple, *Supply Chain*.
7. Red Hat, *Open Source* y Linux Foundation, *Insights*.

Capítulo 9

THE NEW TALENT ARCHITECTURE



Diseñar desde el outcome hacia atrás

Un organigrama puede responder quién reporta a quién. Pregúntele quién produce un outcome de cliente, qué parte depende de un proveedor, dónde está la memoria o quién recupera el sistema después de un fallo. Las cajas empiezan a quedarse en silencio.

No es un defecto del organigrama. Fue diseñado para representar

autoridad formal, no para describir un sistema que mezcla empleo, expertise externo, partners, software e IA. El error consiste en exigirle ambos trabajos.

La **Capability Sourcing Architecture** añade la segunda vista. Parte de un outcome y avanza hacia atrás: capability, workflow, decision rights, modalidad, fuente, interfaz, assurance, memoria y trigger de reconfiguración. No sustituye la estructura legal ni el reporting. Hace visibles dependencias que atraviesan ambos.

Campo 1: Outcome

Todo comienza con un estado deseado para alguien. “Desarrollar producto” es una actividad; “reducir el tiempo desde problema validado hasta release seguro” se aproxima a un outcome. La ficha registra:

- beneficiario;
- estado actual y deseado;
- plazo;
- estándar de calidad;
- restricciones;
- externalidades y daño;
- evidencia de éxito.

El outcome impide diseñar una workforce para mantener ocupadas categorías. También evita que automatización o sourcing se evalúen solo por output local.

Campo 2: Capability

¿Qué debe poder hacer el sistema de manera repetible? Una capability contiene más que skill. Incluye inputs, rutina, acceso, criterio y posibilidad de recuperar. “Saber analizar datos” es habilidad; “detectar una anomalía antes del cierre, explicarla y escalarla” es capacidad más cercana a un outcome.

El registro debe indicar umbral, frecuencia y evidencia. Sin ellos, la capacidad se convierte en un sustantivo aspiracional.

Campo 3: Workflow y dependencias

Dibuje tareas, decisiones, handoffs, excepciones, datos y herramientas. El objetivo no es documentar cada clic, sino localizar interdependencias y cuellos de botella. ¿Qué pieza parece modular y cuál necesita conversación continua? ¿Dónde espera el trabajo? ¿Qué excepción vuelve siempre al founder?

El workflow revela si el problema es falta de capacidad o mala conexión. Contratar a otra persona en una cola bloqueada puede aumentar inventario sin mejorar flow.

Campo 4: Decision rights

Cada outcome necesita al menos:

- un **owner** que prioriza y asigna;
- quien **ejecuta**;
- quien **revisa**;
- un **escalation path**;
- la entidad o persona que **responde**.

Estos papeles pueden coincidir o separarse. Con IA o proveedores

externos, separarlos explícitamente es esencial. Un output puede generarse fuera; la aceptación y accountability permanecen en una institución.

Campo 5: Modalidad y fuente

Aplice BBBA a la unidad que superó el Separability Test. Registre no solo la elección, sino la alternativa. “Borrow mediante partner comprometido” informa más que “vendedor”. “Automate clasificación con revisión humana y fallback manual” informa más que “usar IA”.

La modalidad describe governance; la fuente, quién o qué puede ejecutarla. Cambiar de fuente no siempre cambia de modalidad. Cambiar modalidad puede alterar control, aprendizaje y riesgo.

Campo 6: Interface

¿Qué necesita recibir la fuente para producir? Contexto, datos, permisos, herramientas, estándar, cadencia de feedback y regla de escalado. ¿Qué debe devolver además del output? Evidencia, decisiones, excepciones y conocimiento.

La interfaz no siempre es un documento. En trabajo ambiguo puede ser una relación, rituales o pairing. La arquitectura no fuerza modularidad; revela cuánto contacto necesita la capacidad.

Campo 7: Assurance

¿Cómo sabremos que el resultado cumple antes de causar daño? Defina tests, muestreo, revisión, controles de seguridad, trazabilidad y errores severos. Una capacidad que nadie puede evaluar no está lista para una frontera abierta.

Assurance también debe ser independiente de la fuente cuando el

riesgo lo exige. El proveedor que ejecuta no puede ser la única voz que certifica.

Campo 8: Memory

¿Dónde quedan decisiones, razones, artefactos y excepciones? Linda Argote distingue creación, retención y transferencia de conocimiento organizativo.¹ Una empresa puede producir el mismo output durante meses y no aprender si todo el contexto se marcha con la fuente.

Memory necesita owner y formato útil. Un repositorio lleno no es memoria operativa si nadie puede recuperar la decisión bajo presión.

Campo 9: Fallback y recovery

¿Qué modo degradado existe? ¿Cuánto tarda el reemplazo? ¿Se pueden exportar datos y artefactos? ¿La organización conserva capacidad para diagnosticar? Recovery forma parte de la arquitectura desde el principio, no como anexo de continuidad.

Campo 10: Trigger

Toda configuración es una hipótesis. El trigger indica qué cambio de volumen, calidad, riesgo, coste, mercado o tecnología obliga a revisarla. Sin fecha y señal, una arquitectura temporal se convierte en costumbre.

GitLab: documentación como infraestructura

GitLab es una compañía con empleo, no una prueba de organización “sin personas”. Su interés para esta arquitectura está en el handbook

y en la documentación de prácticas de trabajo all-remote.² Cuando la copresencia no puede transportar contexto informalmente, la organización invierte en artefactos, workflows y reglas visibles.

La documentación reduce algunas fricciones y crea otras. Puede acelerar onboarding, hacer decisiones auditables y permitir contribución asíncrona. También puede volverse obsoleta, ceremonial o abrumadora. El valor no reside en el número de páginas, sino en si una persona puede actuar y decidir mejor.

GitLab ilustra una regla: **la apertura de la ejecución exige mayor claridad en ciertas interfaces, no ausencia de organización**. Lo que el pasillo resolvía de forma informal debe aparecer en otro lugar, aunque parte del conocimiento siga siendo tácito.

NASA: especificar sin vaciar

El NASA Tournament Lab permite externalizar la búsqueda de soluciones para problemas adecuados. La institución conserva la formulación del problema, los criterios, la evaluación y la integración. Esa división puede mapearse en el canvas:

- outcome institucional;
- challenge separable;
- crowd como fuente;
- reglas y artefactos como interfaz;
- expertos para assurance;
- integración y responsabilidad en NASA.

La capacidad externa no reemplaza la capacidad de reconocer una buena respuesta. De hecho, la vuelve más valiosa.

P&G: apertura y absorción

Connect + Develop muestra algo similar en innovación. La empresa puede ampliar el origen de ideas y tecnologías, pero necesita capabilities internas de búsqueda, IP, evaluación, manufactura, marca y adopción. El canvas evita la frase “la innovación viene de fuera” porque obliga a seguir el outcome hasta el mercado.

Los tres casos —GitLab, NASA y P&G— contienen una misma lección en niveles distintos: el acceso funciona sobre infraestructura organizativa. Documentation, challenge design e absorptive capacity son partes de la arquitectura, no overhead posterior.

Contradicciones útiles

El canvas no debe quedar limpio. Su valor aparece cuando marca contradicciones:

- outcome crítico, fuente accesible sin reserva;
- automatización alta, assurance débil;
- proveedor comprometido, memoria sin owner;
- accountable executive, sin derecho de acceso a logs;
- conocimiento estratégico, sin plan de transferencia;
- alta concentración, fallback no probado;
- relación independiente, control operativo incompatible con la etiqueta.

Cada contradicción es una deuda. Puede aceptarse con fecha y rationale; lo que no debe hacer es permanecer invisible.

La objeción burocrática

Diez campos para cada capacidad pueden convertirse en un museo de documentación. Equipos en exploración no conocen el outcome con precisión. La arquitectura podría congelar ambigüedad productiva.

La solución es proporcionalidad. Para una tarea reversible y de bajo daño, bastan outcome, owner, source, interfaz mínima y trigger. Para una capacidad crítica o regulada, se completan los diez. El canvas se utiliza en conversación y decisión, no como requisito de compliance indiferenciado.

Cuando la ambigüedad es real, la conclusión puede ser “necesitamos una relación más estrecha y aprendizaje compartido”. La herramienta no obliga a modularizar lo que todavía no se entiende.

Una arquitectura como hipótesis

Un canvas no mejora performance por existir. Debe predecir algo: menor time-to-ramp, menos retrabajo, mejor recovery, aprendizaje retenido o decisiones más claras. La organización puede comparar tres capacidades piloto y observar si los gaps identificados se relacionan con outcomes.

La arquitectura es exitosa cuando permite decidir y aprender. Fracasa cuando solo renombra organigrama y procurement.

Al terminar esta parte hemos pasado de categoría a sistema. Podemos ver composición, elegir mínimo y conectar outcome con fuente. Pero todavía no sabemos dónde buscar capacidad ni qué profundidad y latencia ofrece cada capa. El siguiente capítulo entra en el **Talent Cloud**, con una advertencia: un mercado grande no es una capacidad grande.

Dos arquitecturas que deben reconciliarse

La organización necesita conservar el organigrama porque autoridad, empleo y responsabilidad formal importan. También necesita un mapa de capabilities porque el trabajo atraviesa esas líneas. El error no consiste en elegir uno. Consiste en permitir que ambos diverjan sin una reconciliación explícita.

Una unidad puede aparecer como owner de un outcome y depender de datos, tecnología y proveedores que gobiernan otras funciones. Un partner puede ejecutar la mayor parte del workflow sin figurar en la estructura. Una persona puede tener autoridad jerárquica y carecer de acceso a los logs con los que debería aceptar el resultado. El **Capability Sourcing Canvas** muestra esas contradicciones y las devuelve al sistema formal.

La reconciliación puede adoptar una matriz sencilla. Cada outcome material tiene un accountable executive en la estructura legal; una capability owner en la operación; decision rights distribuidos; y fuentes con interfaces. Si dos papeles no coinciden, se documenta cómo se resuelve el conflicto. El objetivo no es crear otra jerarquía, sino impedir que la ejecución real quede sin autoridad reconocible.

Diseñar los decision rights

RACI y herramientas similares ayudan a nombrar participantes, pero una arquitectura abierta necesita más precisión sobre decisiones. Conviene distinguir al menos cinco derechos: definir el estándar, asignar recursos, ejecutar, aceptar el output y detener/escalar. En condiciones normales pueden reunirse. En capacidades de alto riesgo, separarlos crea control.

Un proveedor puede ejecutar y proponer cambios, mientras la

empresa conserva aceptación y bloqueo. Un sistema de IA puede actuar dentro de permisos limitados y necesitar confirmación para acciones irreversibles. Una comunidad puede decidir su roadmap y la empresa decidir qué versión integrar. La arquitectura debe reconocer que no toda fuente obedece a la jerarquía interna.

El campo más importante es el de contingencia: ¿quién decide cuando el caso no está previsto? Los contratos y workflows describen lo normal; el derecho residual aparece en la excepción. Si nadie puede responder sin convocar una cadena de reuniones, el diseño ha dejado autoridad en blanco.

La memoria como producto

Cada workflow produce un outcome y una estela de aprendizaje. La arquitectura debe tratar esa estela como un producto con owner. Incluye decisiones, razones, excepciones, feedback, métricas y cambios. No todo necesita convertirse en texto. Parte puede vivir en código, datos, configuraciones, tests o relaciones. Lo importante es que la organización sepa dónde está y quién puede recuperarlo.

Una obligación genérica de “documentar” suele fracasar porque no define usuario. La memoria útil responde a una decisión futura: ¿qué necesitará saber la siguiente persona para operar, evaluar o cambiar? Un log de incidencias sirve para detectar patrones; un registro de decisiones evita reabrir debates; un modelo de datos conserva definiciones; una retrospectiva cambia una interfaz.

La calidad de la arquitectura puede medirse por *memory reuse*: cuántas veces el aprendizaje modifica una especificación, reduce el *ramp time* o mejora acceptance. Un repositorio creciente sin reutilización es almacenamiento, no memoria organizativa.

Arquitectura por fases

La misma capacidad puede necesitar diseños distintos durante exploración, estabilización y escala. En exploración, el outcome es incierto y una relación cercana con expertos puede ser mejor que una interfaz rígida. En estabilización, aparecen patrones y se definen acceptance, datos y roles. En escala, la estandarización y automatización ganan valor; el riesgo de concentración también crece.

El trigger debe incluir el cambio de fase. Una startup que mantiene para siempre una arquitectura de exploración depende de conversaciones y personas clave. Una empresa que intenta escalar demasiado pronto codifica supuestos débiles. La arquitectura no es un blueprint final, sino una serie de configuraciones con criterios de paso.

Una revisión puede preguntar: ¿qué incertidumbre justificaba la configuración actual?, ¿qué hemos aprendido?, ¿qué parte ya puede estandarizarse?, ¿qué parte se ha vuelto estratégica?, ¿qué dependencia apareció? Las respuestas orientan BBBA sin volver a empezar desde cero.

De la planificación anual al ledger vivo

El canvas es más valioso cuando se conecta a un **Capability Ledger**. Cada fila conserva owner, fuentes, decisión, performance, dependencia y trigger. No necesita actualizarse en tiempo real. Sí debe cambiar cuando ocurre una contratación material, un nuevo proveedor, una automatización, un incidente o una revisión estratégica.

El ledger permite observar patrones entre capacidades. Quizá la misma persona integra demasiadas. Quizá todas dependen del

mismo dato. Quizá el tiempo de ramp empeora en fuentes accesible y justifica una red committed. Quizá los triggers nunca se revisan. Esa visión transversal no aparece en un organigrama ni en un sistema de compras.

La arquitectura madura cuando se vuelve parte de decisiones ya existentes: presupuesto, workforce planning, vendor risk, tecnología y continuidad. Crear un proceso paralelo la condena a la obsolescencia. El registro debe ser ligero para capacidades reversibles y profundo para las materiales.

El papel de People y Procurement

Una arquitectura por capacidades no elimina las especialidades funcionales. Las obliga a colaborar sobre el mismo objeto. People aporta diseño de empleo, desarrollo, cultura y derechos. Procurement aporta mercado, contratos, concentración y performance de proveedores. Tecnología aporta interfaces, seguridad y observabilidad. Finanzas aporta coste total y opciones. Operaciones aporta el outcome y las excepciones.

Ninguna función puede sustituir a las demás. Tratar todo como “talento” puede borrar obligaciones laborales. Tratar todo como compra puede borrar aprendizaje y relación. Tratar todo como tecnología puede antropomorfizar sistemas y reducir personas a componentes. La capability es una unidad común, no una categoría administrativa única.

El owner end-to-end reúne esas perspectivas en una decisión. Si la organización descubre que nadie puede desempeñar ese papel, ha encontrado una capacidad incompleta.

Diseño desde el outcome hacia atrás. Reporting lines explican autoridad; una Capability Sourcing Architecture explica cómo el sistema produce, aprende y se recupera.

NOTAS DEL CAPÍTULO

1. Linda Argote, *Organizational Learning: Creating, Retaining and Transferring Knowledge*, Springer.
2. GitLab, [About GitLab](#) y [The Remote Manifesto](#). Fuentes corporativas utilizadas para describir prácticas declaradas, no para probar superioridad.

HERRAMIENTA — PORTFOLIO AND ARCHITECTURE REVIEW

CAPABILITY PORTFOLIO

Construya tres vistas sincronizadas: horas humanas, gasto total y outcomes. Añada concentración, covarianza, fallback, aprendizaje y riesgo bilateral.

SOLID MINIMUM TEST

Para cada capacidad crítica declare el objeto mínimo de control bajo Advantage, Residual control, Cumulative learning, System accountability, Bottleneck y Complementarity. Registre también el coste de rigidez.

CAPABILITY SOURCING CANVAS

Complete outcome → capability → workflow → decision rights → modalidad/fuente → interface → assurance → memory/recovery → trigger. Señale contradicciones; no las oculte con una nota media.

Output de la parte: tres deudas priorizadas —concentración, memoria y accountability— y una decisión con owner y fecha.

El día en que desapareció la capacidad

Un lunes, una fuente crítica deja de estar disponible. Puede ser una persona, un proveedor, una plataforma, un repositorio, un modelo o una jurisdicción. La causa importa menos que la secuencia. El equipo descubre que los contratos están vigentes, las facturas al día y el portfolio en verde. El outcome se detiene.

La primera hora revela quién conoce el sistema. El primer día revela quién puede decidir. La primera semana revela qué memoria permanece. El primer mes revela si la capacidad pertenecía a la organización o solo al estado anterior del mundo.

Este escenario es la prueba conjunta de portfolio, Solid Minimum y architecture.

Hora cero: localizar el outcome

Los equipos suelen empezar por la fuente: llamar al proveedor, sustituir a la persona, buscar otra herramienta. El portfolio maduro empieza por el outcome. ¿Qué promesa está en riesgo? ¿Qué modo degradado es aceptable? ¿Qué decisiones deben tomarse antes de buscar reemplazo?

Esta orientación evita recrear exactamente la configuración que falló. Quizá el workflow puede simplificarse, el volumen priorizarse o una tarea suspenderse. Recovery no significa volver a la misma arquitectura; significa restaurar capacidad suficiente.

El accountable owner necesita información transversal. Si gasto, permisos, contratos y memoria están en sistemas distintos, la

arquitectura administrativa se convierte en parte del incidente.

Día uno: descubrir concentración

La fuente desaparecida puede haber sido solo el primer nivel. Al buscar alternativas aparecen common-mode dependencies. Todos los especialistas disponibles utilizan la misma plataforma. El segundo proveedor necesita datos que solo gestionaba el primero. La persona interna de respaldo no ha practicado. El integrador también está ausente.

La diversificación contractual se deshace. El portfolio debe reconstruirse por dependencias: infraestructura, dato, conocimiento, geografía, decisión y relación. Una alternativa es independiente solo si sobrevive al escenario.

La empresa quizá decida operar con mayor coste o menor calidad durante la transición. Esa elección debe estar preautorizada en el *recovery minimum*. Esperar perfección puede alargar el daño.

Semana uno: la memoria en acción

Los repositorios contienen documentos y nadie sabe cuál refleja la práctica. Las decisiones importantes quedaron en reuniones. Un contrato exige transferencia y no define formato. La organización posee archivos y carece de memoria.

La memoria operativa se reconoce porque permite actuar: identificar excepciones, reconstruir criterios, contactar a stakeholders y explicar por qué una regla existe. Su owner no tiene que conocer todo, pero debe poder reunirlos.

Si la salida planificada nunca se ensayó, el incidente es la primera prueba. El coste resultante pertenece a Total Access Cost aunque

aparezca años después de seleccionar la fuente.

Mes uno: elegir un nuevo mínimo

La empresa puede internalizar, comprometer una fuente nueva, repartir el workflow o automatizar. El instinto posterior al fallo es aumentar control. Puede ser correcto y también excesivo. El Solid Minimum Test pregunta qué propiedad faltó: juicio, derechos, redundancia, memoria, assurance o recovery.

Si faltó portabilidad, comprar un activo completo puede ser innecesario. Si faltó criterio interno, otro proveedor no resuelve. Si el mercado falló por shock común, una reserva sobre la misma oferta no sirve. Si la capacidad era demasiado interdependiente, el bundle debe cambiar.

El mínimo se diseña contra el mecanismo del fallo, no contra el miedo general.

El balance bilateral del shock

Durante una crisis, las empresas pueden trasladar urgencia hacia contractors y proveedores: plazos extraordinarios, disponibilidad no reservada, cambios de scope. La recuperación aparente se financia con jornadas y riesgo fuera del dashboard.

Una arquitectura responsable distingue emergencia real de mala planificación, remunera reserva y esfuerzo, y protege límites de seguridad. El shock no suspende la bilateralidad; la hace más material.

También puede afectar a empleados que absorben toda integración. Contar horas extraordinarias y carga emocional evita declarar resiliente a un sistema que se recupera quemando a su

núcleo.

Después del incidente

Una retrospectiva útil actualiza el Capability Ledger. Registra qué dependencia no se vio, qué trigger llegó tarde, qué interfaz falló, qué memoria fue utilizable y quién absorbió coste. Después modifica arquitectura, no solo el plan de continuidad.

El portfolio puede aceptar la misma exposición con un rationale distinto. Puede financiar una opción, cambiar proveedor, distribuir conocimiento o reducir complejidad. No toda lección exige redundancia.

La prueba final ocurre meses después: ¿el siguiente onboarding es más rápido?, ¿el fallback funciona?, ¿el owner puede explicar?, ¿la carga de integración bajó? Si no, el incidente produjo un informe y no aprendizaje.

Una capacidad, cuatro configuraciones

Considere una capacidad común: producir un informe mensual de rendimiento comercial que permita decidir precios, inversión y prioridades. El outcome exige datos correctos, interpretación, explicación de desviaciones y una decisión a tiempo. La empresa puede configurarlo de cuatro maneras.

La primera desarrolla y ejecuta todo dentro. La segunda incorpora de forma estable a una persona que posee el ciclo. La tercera utiliza una firma o profesional recurrente. La cuarta automatiza extracción, análisis y borrador, con revisión humana. En la práctica habrá híbridos; separarlas ayuda a observar trade-offs.

Configuración Build

La empresa desarrolla skills de análisis en su equipo actual, crea el modelo de datos y distribuye responsabilidades. La ventaja potencial es aprendizaje integrado: quienes conocen clientes y producto construyen interpretación. El coste es tiempo, posible falta de expertise y distracción de otras prioridades.

Build funciona mejor si el conocimiento es estratégico, existe base interna y la demanda permite aprender. Falla si se confunde construir capacidad con asignar trabajo adicional sin slack. También puede reinventar métodos disponibles.

El trigger puede ser incapacidad de alcanzar calidad tras varios ciclos, crecimiento de volumen o coste de oportunidad. El fallback

puede ser asesoramiento externo mientras el equipo mantiene ownership.

Configuración Buy

La empresa contrata a una persona para poseer datos comerciales y reporting. Compra continuidad, disponibilidad, accountability y una opción para resolver excepciones. Puede desarrollar una función que más adelante abarque forecasting y pricing.

El riesgo es diseñar un puesto alrededor de un informe mensual. Si el volumen no llena el bundle, la organización añadirá tareas heterogéneas y puede perder foco. También puede contratar antes de arreglar datos, convirtiendo a la persona en integrador manual.

Buy encaja si la demanda es continua, el aprendizaje acumulativo importa y existen decision rights reales. El trigger debe observar utilización de alto valor, no horas ocupadas. Si la persona pasa la mayor parte del tiempo limpiando inputs que otras áreas no gobiernan, el problema es architecture.

Configuración Borrow

Una firma recurrente recibe datos, prepara análisis y participa en la revisión. Aporta métodos, benchmarks y flexibilidad. La empresa conserva owner de decisión y criterios de aceptación.

Borrow puede acelerar y ofrecer expertise que no justifica un puesto. El coste aparece en contexto, iteración y dependencia. Si la firma explica cada mes las mismas desviaciones y nadie interno aprende, la capacidad permanece fuera. Si la relación rota personal, el compromiso nominal no garantiza memoria.

La interfaz necesita definiciones de datos, calendario, hipótesis,

decision rights y transferencia. El trigger puede ser aumento de especificidad, concentración o demanda suficiente para internalizar. Una salida ensayada conserva modelos y razones.

Configuración Automate

Software extrae datos, calcula indicadores, detecta anomalías y genera un borrador. Una persona revisa, investiga excepciones y decide. La automatización puede reducir trabajo repetitivo y mejorar puntualidad. No conoce por sí sola qué cambio de mercado vuelve engañosa una métrica.

El análisis debe contar preparación de datos, mantenimiento, review y severe errors. Si el sistema ahorra redacción y aumenta investigación de falsos positivos, el beneficio es diferente del esperado. Si permite que el owner dedique más tiempo a decisiones y los errores están controlados, puede ampliar capacidad.

El trigger incluye cambios de modelo, drift de datos, tasa de override y pérdida de skill. El fallback es un modo manual o un informe reducido que la empresa puede producir sin el proveedor técnico.

La combinación

Una arquitectura plausible construye definiciones y decision rights dentro, utiliza un especialista committed para diseñar y revisar durante una fase, automatiza extracción y mantiene a una persona accountable. Con el tiempo puede internalizar más análisis o reducir la relación externa.

La combinación no es superior por ser híbrida. Añade interfaces. Debe demostrar que cada pieza aporta algo: conocimiento, opción,

escala o independencia de assurance. Si la persona interna, el proveedor y la herramienta revisan lo mismo, la pluralidad es duplicación.

La comparación simétrica

Para las cuatro configuraciones se registran los mismos campos: time-to-capability, coste directo, horas de management, first-pass acceptance, error severo, aprendizaje, disponibilidad, switching y condiciones bilaterales. No se compara salario con factura ni factura con licencia.

Después de seis meses, la empresa observa el outcome. Quizá Borrow fue más rápido y Build generó mejor aprendizaje. Quizá Automate redujo latencia y necesitó más assurance. Quizá Buy convirtió un proceso inestable en función estable. La decisión estratégica depende de qué propiedad es más valiosa y de qué opción quiere preservar.

La lección

La misma capacidad puede producirse mediante configuraciones distintas y ninguna existe en abstracto. Cada una contiene relaciones, derechos, trabajo invisible y una trayectoria de aprendizaje. BBBA no responde con una letra. Obliga a dibujar el sistema, elegir una hipótesis y aprender de la comparación.

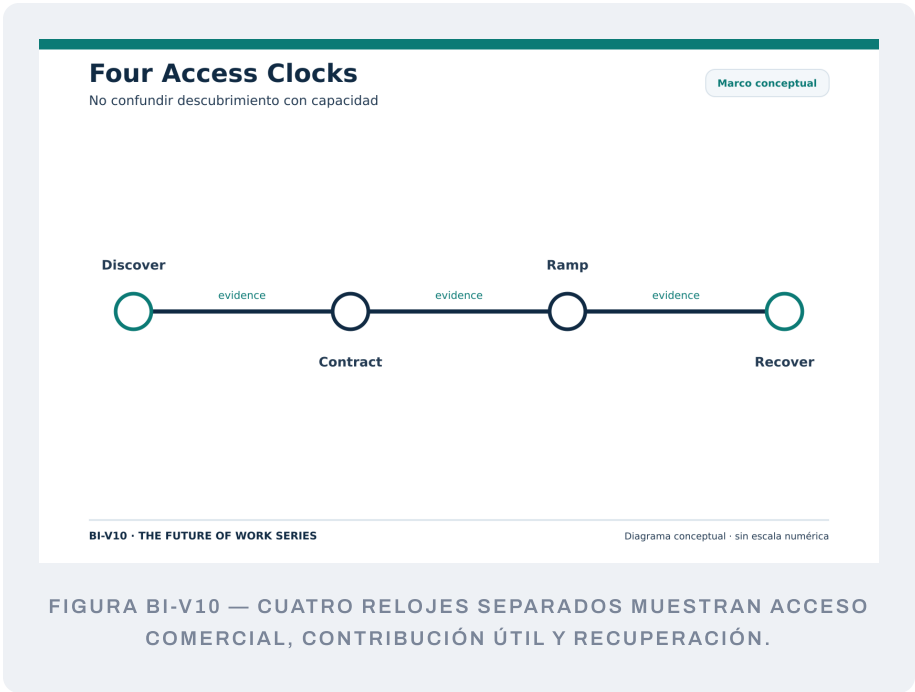
PARTE IV

DE LA CAPTACIÓN A LA ORQUESTACIÓN DEL TALENTO

Encontrar capacidad no es convertirla en resultado. Esta parte estudia el impuesto de integración y las rutinas que hacen posible descubrir, contextualizar, coordinar, evaluar, aprender y recombinar.

Capítulo 10

THE TALENT CLOUD



Profundidad, latencia y compromiso

Dos empresas pueden abrir la misma plataforma y ver el mismo especialista. Una recibe respuesta, acuerda scope y comienza en días. La otra publica un brief ambiguo, tarda en aprobar presupuesto, no puede conceder permisos y concluye que “el mercado no funciona”. El tamaño del marketplace no ha cambiado. La capacidad de acceso de la empresa, sí.

No tenemos un dataset que permita narrar esas dos empresas

como caso identificado; la comparación es un mecanismo. Evita una confusión frecuente: **oferta visible no es capacidad movilizable**.

Thomas Malone y Robert Laubacher imaginaron en 1998 una economía de e-lancers conectados electrónicamente.¹ El antecedente importa. El *human cloud* no nació con la IA generativa ni con la última generación de marketplaces. Lo que ha cambiado es escala, infraestructura y normalización. Las fricciones, sin embargo, no han desaparecido.

Tres Access Layers

El título *Talent Cloud* es una metáfora comercialmente potente. El modelo operativo es más preciso: **Access Layers**.

CONTROLLED CORE

Capacidades bajo control estable. Ofrecen mayor continuidad y contexto, aunque payroll no garantiza que la skill correcta esté disponible. Un core puede estar saturado, desactualizado o mal conectado.

COMMITTED NETWORK

Relaciones externas con reserva, repetición, reputación y memoria. Incluye especialistas recurrentes, partners, acuerdos marco y managed services con equipo identificado. Reduce time-to-access a cambio de compromiso, coste de reserva o dependencia.

MARKET / OPEN SET

Marketplaces, concursos, comunidades, expert networks y oferta abierta. Maximizan variedad potencial. Exigen especificación, vetting, bidding, permisos y mayor tolerancia a latencia o variación.

Las capas no representan mejor/peor. Se diferencian por profundidad, tiempo, compromiso, contexto y riesgo.

No todos los mercados compran lo mismo

Un marketplace abierto permite comparar perfiles y propuestas. Una red curada restringe oferta y añade selección. Un managed service asume parte de la coordinación. Un concurso busca la mejor solución a un problema. Una comunidad contribuye bajo sus propias motivaciones y governance. Una expert network vende conversaciones o acceso a conocimiento.

Llamarlos a todos “talent platforms” oculta mecanismos:

- ¿se compra tiempo, output, solución o relación?;
- ¿quién selecciona?;
- ¿cómo se forma reputación?;
- ¿quién asume disputa y pago?;
- ¿hay continuidad?;
- ¿qué control ejerce el intermediario?;
- ¿cómo sale cada parte?

Upwork y Fiverr publican filings que describen modelos de plataforma, riesgos, ingresos y cambios tecnológicos.² Esas fuentes permiten entender el intermediario. No nos dicen por sí solas cuánto tarda una empresa concreta en lograr productividad ni qué experiencia tiene cada profesional. El dossier correcto combina filing, datos de uso y perspectiva bilateral.

Profundidad real

La profundidad de una capa no es el número registrado en una web. Es un funnel:

1. oferta visible;
2. cualificación relevante;
3. disponibilidad en el periodo;
4. autorización jurídica y de datos;
5. economics aceptables;
6. capacidad de integrarse;
7. probabilidad de cumplir y permanecer;

La cifra se reduce en cada paso. Para una capacidad escasa, la profundidad efectiva puede ser una sola persona comprometida. Para una tarea estandarizada, un mercado abierto puede mantener mucha oferta.

Cuatro relojes

El acceso contiene cuatro tiempos que suelen colapsarse en “time-to-hire”.

Search time: desde la necesidad hasta encontrar candidatos.

Access time: hasta acordar condiciones y disponibilidad.

Ramp time: hasta comprender contexto, herramientas y estándares.

Capability time: hasta producir repetidamente por encima del umbral y recuperarse de errores.

Una plataforma puede reducir search y access sin cambiar ramp. Una relación comprometida puede parecer más lenta de construir y

ser mucho más rápida de activar. El reloj correcto depende del outcome.

Persistencia y reputación

La visión de un mercado spot supone transacciones discretas. La evidencia sobre plataformas online muestra que la persistencia relacional importa. Stanton y Thomas encuentran mecanismos por los que intermediarios y reputación afectan quién se beneficia, y las relaciones repetidas no desaparecen.³

Esto explica por qué el Committed Network puede ser la capa más infravalorada. Las empresas construyen un pequeño mercado privado: personas que ya conocen el contexto, han demostrado fiabilidad y pueden reservarse con mayor rapidez. No tiene la variedad total del open set ni el control del core. Compra una combinación de opción y memoria.

También crea switching cost. Si toda la capacidad vive en tres relaciones, la red puede ser robusta en el día a día y frágil ante salida. El Access Layers Map debe superponer concentración y transferibilidad.

La empresa también compite

Hablar de “acceso al mejor talento” suena unilateral. Los especialistas eligen. Evalúan claridad, propósito, pago, reputación, autonomía, comportamiento y probabilidad de cobrar. Un brief confuso, un proceso lento o un historial de scope creep reduce la profundidad real.

La marca compradora es una capacidad. Incluye:

- especificaciones honestas;
- decisiones rápidas;
- precio y plazo de pago;
- respeto de autonomía;
- acceso suficiente sin vigilancia excesiva;
- feedback y reconocimiento;
- posibilidad de relación repetida;
- salida limpia.

Una empresa no “tiene” un Talent Cloud. Participa en mercados y relaciones en los que otras partes también gobiernan el acceso.

Global en catálogo, local en obligación

Una plataforma puede mostrar oferta mundial. La ejecución sigue atravesando idioma, huso, pagos, propiedad intelectual, privacidad, export controls, impuestos y clasificación laboral. Esas restricciones no son fricción accidental; protegen derechos y distribuyen responsabilidad.

El Access Layers Map debe llevar geografía. Una fuente puede ser accesible para diseño no sensible y estar fuera de alcance para datos regulados. Una persona puede trabajar remotamente y crear obligaciones para empresa o profesional. “Global” describe el espacio de búsqueda, no una exención institucional.

Condiciones humanas como parte del acceso

La metáfora cloud puede volver invisibles a quienes trabajan. Compute parece instantáneo; expertise no lo es. Una persona

mantiene carrera, límites, aprendizaje y múltiples compromisos. El plazo de pago, el control, el tiempo no facturado y la posibilidad de apelación forman parte de la calidad del mercado.

La OIT ha subrayado oportunidades y déficits de trabajo decente en plataformas digitales.⁴ Incorporar estas variables no es añadir una sección ética después del modelo. Es medir la sostenibilidad de la fuente. Un mercado que quema oferta o depende de trabajo no remunerado puede parecer profundo antes de deteriorarse.

Access Layers Map

Para una capacidad escasa, registre por capa:

VARIABLE	MEDIDA O EVIDENCIA
Profundidad elegible	fuentes que pasan cualificación, disponibilidad y permiso
Time-to-access	mediana y cola, no solo mejor caso
Time-to-ramp	hasta primer outcome aceptado
Commitment	reserva, tenure, SLA y prioridad
Variación	calidad, precio y disponibilidad
Total Access Cost	integración, review, error y salida
Bilateral conditions	pago, control, gastos, voz y dependencia
Constraint	qué rompe primero la capa

No sume las capas como si fueran inventario. El core puede habilitar al network; el network puede seleccionar del market. El mapa muestra rutas, no stock.

La objeción del viejo procurement

¿No es esto vendor management con nombres nuevos? En parte comparte problemas antiguos: sourcing, acuerdos, performance y riesgo. La diferencia útil es que la unidad se amplía desde proveedor hasta capability, incluye empleo y automatización, separa catálogo de frontera efectiva y mide aprendizaje y bilateralidad. Si una organización ya hace todo eso, no necesita la etiqueta.

El capítulo no reclama que encontrar talento sea nuevo. Reclama que el coste de búsqueda ha cambiado para algunas capacidades y que la ventaja se desplaza hacia cómo una empresa convierte fuente en outcome.

El Talent Cloud resuelve una parte: **dónde buscar y con qué profundidad**. El trabajo decisivo comienza después. Especificar, integrar, asegurar, aprender y reconfigurar puede costar más que el match. Ese conjunto de rutinas es **Talent Orchestration**.

El comprador también tiene un *access stack*

Las plataformas describen la oferta; la empresa necesita describir su lado de la conexión. Un **buyer access stack** contiene presupuesto autorizado, capacidad para especificar, selección, contrato, identidad, permisos, datos, herramientas, onboarding, acceptance, pago y salida. La ausencia de una sola capa puede convertir mercado profundo en acceso lento.

Este stack explica diferencias entre empresas que utilizan la misma fuente. Una puede emitir un brief claro, decidir en días y conceder accesos proporcionados. Otra fragmenta aprobación entre siete áreas, no sabe qué resultado aceptar y paga a noventa días. El problema no es la nube de talento. Es una arquitectura compradora que destruye la opción que pretende adquirir.

La mejora más rentable puede estar dentro: plantillas de capability brief, acuerdos preaprobados, roles de acceso, entornos seguros, criterios de acceptance y procesos de pago. No son tareas administrativas periféricas. Son infraestructura que reduce *access* y *ramp time* sin prometer que la fuente externa sea superior.

Niveles de servicio para el acceso

Las capas pueden definirse mediante niveles de servicio. Para una capacidad concreta, la empresa declara qué puede activar en veinticuatro horas, setenta y dos horas, treinta días y un trimestre. Cada horizonte corresponde a una profundidad distinta.

En veinticuatro horas quizá solo exista el core y una guardia committed. En setenta y dos, una red de especialistas con accesos preparados. En treinta días, selección de mercado y onboarding. En un trimestre, formación, contratación o construcción. El mapa evita prometer “on demand” cuando la capacidad necesita tiempo institucional.

Los horizontes también revelan huecos. Si una función crítica solo puede cubrirse en treinta días y el daño empieza en horas, el problema no se resuelve aumentando perfiles visibles. Requiere reserva, automatización segura, redundancia o control estable. Si una tarea tolera un mes, pagar por disponibilidad inmediata puede ser desperdicio.

Profundidad de mercado frente a profundidad de relación

Un mercado ofrece profundidad cuando existen muchas fuentes cualificadas y sustituibles. Una relación ofrece profundidad cuando acumula contexto, confianza y capacidad de adaptarse. Las dos

profundidades no son equivalentes. El mercado reduce dependencia de una fuente; la relación reduce latencia e integración.

La arquitectura puede combinar ambas: mantener un committed network y renovar parte de él desde el open set; usar el mercado para explorar y las relaciones para operar; reservar una fuente principal y conservar estándares que permitan segunda fuente. El equilibrio cambia con especificidad y frecuencia.

Una empresa puede equivocarse en ambos sentidos. Convertir cada colaboración exitosa en relación exclusiva reduce variedad. Mantener todo spot obliga a pagar repetidamente por contexto. El portfolio debe observar cuándo la repetición ya justifica compromiso y cuándo el compromiso ha creado captura.

La reputación es bidireccional

En mercados digitales, la reputación suele representarse como una puntuación del profesional. La empresa también acumula reputación, aunque no siempre sea visible en la interfaz. Especialistas comparten información sobre claridad, trato, scope, decisiones y pago. Una organización que consume tiempo en procesos fallidos o cambia condiciones verá reducirse la oferta realmente interesada.

La reputación compradora puede medirse con señales internas: tasa de aceptación de invitaciones, tiempo desde propuesta a decisión, porcentaje de proyectos que amplían scope sin ajuste, plazo real de pago, repetición voluntaria y motivos de salida. Estas medidas no garantizan una relación justa, pero convierten un supuesto unilateral de abundancia en evidencia.

El acceso sostenible requiere que la oportunidad sea valiosa para

ambas partes. El mejor talento no necesita trabajar *para* la empresa y tampoco necesita trabajar *con* ella. La capacidad de atraerlo depende de la calidad institucional del comprador.

Access debt

Una empresa acumula **access debt** cuando gana velocidad hoy mediante excepciones que dificultarán la siguiente conexión. Concede permisos manuales sin inventario, acepta formatos incompatibles, deja decisiones en mensajes privados, depende de una persona para onboarding o contrata sin una ruta de salida. Cada atajo parece pequeño; juntos elevan el *ramp time* y el riesgo.

La deuda puede observarse en síntomas: nuevas fuentes necesitan más reuniones, los owners no saben qué accesos existen, el mismo contexto se explica repetidamente, los artefactos no son reutilizables y finalizar una relación tarda más que iniciarla. La solución no es documentar todo, sino reparar las interfaces de mayor frecuencia y daño.

Una revisión después de cada colaboración importante puede preguntar qué parte del onboarding debería convertirse en infraestructura, qué permiso debe estandarizarse, qué decisión debe registrarse y qué dependencia debe eliminarse. Así cada acceso mejora el siguiente en lugar de consumirlo.

Del mercado a la capacidad en cuatro conversiones

Una fuente externa atraviesa cuatro conversiones. De **perfil a compromiso**: selección, disponibilidad y términos. De **compromiso a productividad**: contexto, herramientas y relación. De **productividad a capacidad**: repetición, assurance y recuperación. De **capacidad a aprendizaje**: memoria que cambia el sistema.

Las plataformas suelen optimizar la primera. La empresa es responsable de las demás, aunque un managed service ayude. Medir solo tiempo de contratación premia el match y omite el trabajo que determina el outcome.

Este ciclo permite comparar capas sin moralizarlas. El core puede comenzar más cerca de productividad y estar saturado. La red committed puede comenzar con contexto y tener menor variedad. El market puede encontrar skills raras y exigir más conversión. El diseño correcto usa cada una donde su perfil encaja.

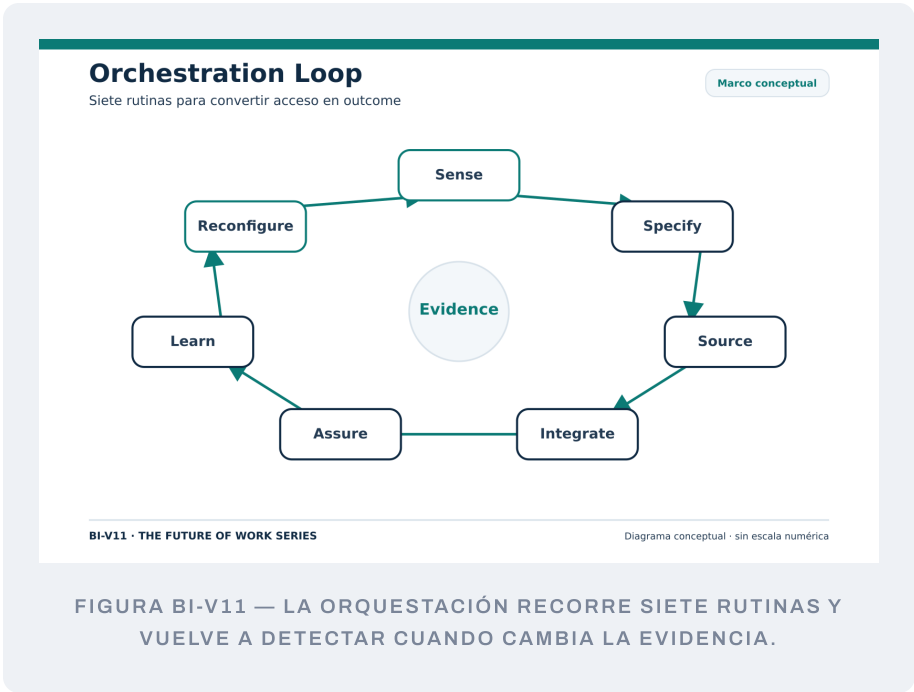
No mida su Talent Cloud por perfiles visibles. Mídalo por fuentes cualificadas, disponibles, integrables y sostenibles, y por el tiempo hasta capacidad repetible.

NOTAS DEL CAPÍTULO

1. Thomas W. Malone y Robert J. Laubacher, “**The Dawn of the E-Lance Economy**”, *Harvard Business Review*, 1998.
2. Upwork, **Annual Reports**; Fiverr, **2025 Form 20-F**.
3. Christopher Stanton y Catherine Thomas, “**Who Benefits from Online Gig Economy Platforms?**”, *American Economic Review*, 2025.
4. OIT, **Digital labour platforms**.

Capítulo 11

TALENT ORCHESTRATION



El trabajo después del acceso

Li & Fung se convirtió durante años en un emblema de coordinación de producción distribuida. La compañía no necesitaba poseer cada fábrica para configurar redes de suppliers, información y logística. Su historia posterior —reestructuración, presión competitiva y pérdida de posición— importa tanto como su auge.¹ Una red no concede ventaja permanente. La capacidad para configurarla puede deteriorarse cuando cambian tecnología, clientes o economics.

Ese arco evita convertir *orchestration* en elogio. Orquestar no significa estar en el centro de muchas relaciones. Significa ejecutar rutinas que convierten fuentes dispersas en outcomes fiables, aprendizaje y capacidad de cambio. Si las rutinas no producen algo observable, la palabra sobra.

Acquisition termina demasiado pronto

Talent acquisition celebra cuando la relación comienza.

Procurement celebra al firmar. Una plataforma celebra el match. El cliente, sin embargo, recibe valor después: cuando la fuente entiende, accede, produce, supera assurance, aprende y puede ser sustituida o recombinada.

La distancia entre acceso y outcome contiene trabajo:

- formular necesidad;
- seleccionar modalidad y fuente;
- transferir contexto;
- conceder permisos;
- coordinar dependencias;
- evaluar calidad;
- capturar conocimiento;
- cambiar el arreglo.

Ese trabajo puede recaer en managers, operations, HR, procurement, security y founders. Como está fragmentado, rara vez se contabiliza. Una empresa parece ahorrar capacidad en ejecución y consume capacidad senior en integración.

Los antecedentes

El concepto no nace aquí. Robert Grant describió la firma como integradora de conocimiento especializado.² La literatura de **resource orchestration** estudia cómo estructurar, agrupar y aprovechar recursos.³ Las **dynamic capabilities** de Teece, Pisano y Shuen se ocupan de integrar, construir y reconfigurar competencias ante entornos cambiantes.⁴

La contribución de este libro es operacionalizar esas tradiciones para un sistema que combina capacidades controladas, comprometidas, accesibles y automatizadas. El **Orchestration Loop** tiene siete rutinas: sense, specify, source, integrate, assure, learn y reconfigure.

No son una teoría rival. Son una hipótesis práctica: organizaciones que ejecutan mejor estas rutinas deberían lograr menor time-to-capability, más fiabilidad, mejor recuperación y más aprendizaje, manteniendo constantes otras condiciones. Si la hipótesis no se sostiene, el loop debe quedarse como checklist.

1. Sense

Detectar una brecha de capacidad exige distinguir síntomas de outcomes. “El equipo está saturado” puede ser falta de capacidad, una prioridad incoherente o un workflow bloqueado.

Output mínimo: baseline, outcome, gap, urgencia y coste de no actuar.

Fallo típico: convertir una solicitud de puesto o herramienta en definición del problema.

Medida: tiempo desde señal hasta gap validado; tasa de decisiones

que cambian al analizar el baseline.

2. Specify

La organización traduce el gap en una capability con constraints.

Define calidad, latencia, daño, separabilidad, mínimo sólido y lo que no puede romperse.

Output mínimo: capability brief y acceptance criteria.

Fallo típico: especificación de solución (“necesitamos un vendor”) o un brief tan detallado que impide expertise.

Medida: aclaraciones, scope changes, discrepancias de aceptación y tiempo perdido por ambigüedad.

NASA Tournament Lab ilustra el valor de esta rutina.

Crowdsourcing funciona mejor para problemas que pueden formularse y evaluarse. La comunidad no elimina la capacidad institucional de plantear la pregunta.

3. Source

Elegir modalidad, fuente y alternativa. Source incluye BBBA, Access Layers, due diligence y opción preservada.

Output mínimo: Decision Record, source elegida, alternativa y trigger.

Fallo típico: seleccionar por familiaridad, tarifa o disponibilidad inmediata sin coste total.

Medida: profundidad elegible, time-to-access, concentración y diferencia entre coste previsto y total.

P&G Connect + Develop muestra que buscar fuera puede

convertirse en rutina, no evento. La búsqueda externa descansa sobre capacidad interna de evaluar y adoptar.

4. Integrate

Contexto, permisos, herramientas, ritmo, roles y decision rights convierten una fuente en participante del workflow.

Output mínimo: Capability Interface Contract proporcional al riesgo.

Fallo típico: onboarding administrativo sin contexto; acceso excesivo; founder traduciendo entre piezas.

Medida: time-to-ramp, preguntas recurrentes, handoffs, permisos excepcionales y horas de management.

La integración es el impuesto oculto del acceso. Puede reducirse mediante documentación y estándares, pero no desaparecer cuando el trabajo depende de juicio y relación.

5. Assure

Observar, evaluar, aceptar, bloquear y escalar. Assurance define quién revisa y qué errores severos importan.

Output mínimo: tests, reviewer, logging, escalation y fallback.

Fallo típico: el executor se autocertifica; métricas fáciles sustituyen calidad; revisión nominal sin tiempo ni señal.

Medida: first-pass acceptance, defectos, severidad, overrides, retrabajo y time-to-detection.

Con IA, esta rutina es inseparable del diseño. Un modelo puede producir miles de outputs; si review no escala, la capacidad efectiva

puede reducirse.

6. Learn

Retener decisiones, feedback, excepciones y artefactos. Learning cambia la práctica; archivar no basta.

Output mínimo: decision log, conocimiento transferido, actualización de estándar o cambio de formación.

Fallo típico: la fuente mejora mientras la empresa sigue siendo incapaz de evaluar; lessons learned que nadie reutiliza.

Medida: repetición de errores, time-to-ramp de la siguiente fuente, reutilización y reducción de dependencia.

Linda Argote recuerda que conocimiento organizativo debe crearse, retenerse y transferirse. Una relación exitosa puede no aumentar capacidad si no deja memoria.

7. Reconfigure

Profundizar, diversificar, automatizar, internalizar, simplificar o parar. Reconfiguration cierra el loop porque la configuración inicial es una hipótesis.

Output mínimo: decisión fechada, trigger observado, nueva arquitectura y plan de transición.

Fallo típico: renovación por inercia; escalar un piloto sin revisar daño; mantener un proveedor porque cambiar resulta incómodo.

Medida: time-to-change, switching loss, recovery, calidad posterior y opciones preservadas.

El Solid Minimum en el centro

El loop podría optimizar velocidad y destruir lo esencial. Por eso el Solid Minimum atraviesa integración, assurance y reconfiguration. Cada cambio pregunta: ¿qué debe permanecer para conservar decisión, aprendizaje, recovery y accountability?

Una empresa verdaderamente reconfigurable no es la que cambia más. Es la que puede cambiar sin perder la capacidad de gobernar.

Capability debt

Cuando una organización accede a capacidad sin construir las rutinas necesarias, acumula **capability debt**:

- knowledge debt: nadie retiene razones y excepciones;
- interface debt: cada incorporación requiere improvisación;
- assurance debt: calidad se descubre tarde;
- concentration debt: la sustitución es contractualmente posible y operativamente inviable;
- accountability debt: el owner existe en el organigrama, no en la práctica;
- orchestration debt: el founder o manager absorbe integración no medida.

La deuda puede ser racional durante un experimento. Debe llevar interés, owner y fecha. El problema es convertir un workaround temporal en arquitectura permanente.

Medir al orquestador

La orquestación suele presentarse como leverage: una persona

moviliza muchas capacidades. Necesitamos contar también su carga.

Registre:

- horas de especificación y selección;
- horas de onboarding y contexto;
- decisiones y escalados;
- review y retrabajo;
- mantenimiento de relación;
- documentación y transferencia;
- incidentes y recovery.

Si estas horas crecen linealmente con cada fuente, el modelo no escala. Si estándares, relaciones y aprendizaje reducen el coste marginal, puede estar emergiendo una capacidad de segundo orden.

El **Orchestration Load Ratio** —horas del núcleo dedicadas a integrar y gobernar dividido por horas humanas totales de la capacidad— es una medida exploratoria, no benchmark. Sirve para comparar una misma capacidad a lo largo del tiempo, no para ordenar empresas.

La mejor objeción

Todo esto puede ser simplemente management competente. Managers siempre han detectado necesidades, especificado trabajo, seleccionado personas, coordinado, evaluado y aprendido. ¿Qué añade *Talent Orchestration*?

La respuesta honesta es: quizá solo una operacionalización útil para una arquitectura más plural. Para reclamar más necesitamos

demostrar validez discriminante. El loop debería predecir outcomes después de controlar por calidad de recursos, presupuesto, experiencia directiva y digital maturity. También debería fallar de maneras específicas: buena selección con mala integración; buena ejecución sin aprendizaje; alta velocidad con recovery débil.

Hasta contar con esa evidencia, el claim ceiling es modesto. No diga “orchestration es la nueva ventaja competitiva” como hecho. Diga: **cuando el acceso se amplía, las rutinas de integración, assurance, aprendizaje y reconfiguration se vuelven candidatas a ventaja; su valor debe observarse en outcomes.**

Un audit de fallo

La forma más útil de empezar no es puntuar madurez. Elija una capacidad que falló y recorra el loop:

- ¿Sense confundió síntoma y gap?
- ¿Specify omitió una excepción?
- ¿Source eligió una opción nominal?
- ¿Integrate dejó contexto o permisos sin resolver?
- ¿Assure detectó tarde?
- ¿Learn retuvo algo?
- ¿Reconfigure ocurrió o se renovó por inercia?

El audit localiza trabajo, no culpa. Una fuente puede haber ejecutado correctamente dentro de una arquitectura imposible.

El loop formaliza procesos. La colaboración, sin embargo, depende también de legitimidad, voz, confianza y conflicto. Una empresa no puede dirigir a toda fuente mediante la misma

autoridad. El capítulo siguiente aborda la pregunta que los contratos no resuelven: **¿cómo liderar cuando no existe una jerarquía común?**

Orquestación distribuida, no un nuevo centro

El lenguaje de orquesta sugiere una persona que dirige todos los instrumentos. Esa imagen puede ser engañosa. En una organización compleja, ninguna figura posee suficiente contexto para especificar, integrar y asegurar cada capacidad. Centralizar la orquestación produciría el mismo cuello de botella que el framework intenta revelar.

Las siete rutinas deben distribuirse con ownership claro. Operaciones puede *sense* el gap; un capability owner especifica; People, Procurement o Tecnología ayudan a source; seguridad y responsables de sistemas integran permisos; expertos de dominio aseguran; el equipo retiene aprendizaje; un foro de portfolio decide reconfigurar. La distribución no elimina un owner end-to-end. Evita que ese owner realice personalmente todo el trabajo.

La coordinación necesita un protocolo común: artefactos mínimos, decision rights, escalado y métricas. Así una capacidad de bajo riesgo puede fluir de manera autónoma y una excepción material puede recibir atención. La orquestación madura no se reconoce porque muchas decisiones llegan al centro, sino porque las decisiones correctas ocurren cerca del conocimiento y siguen siendo visibles para quien responde.

El cuello de botella cambia de lugar

Acceder a más ejecución puede liberar capacidad operativa y concentrar especificación, revisión o decisión. Cuando una empresa conecta IA, proveedores y especialistas, el nuevo recurso escaso

suele ser atención de quien posee contexto. Ese desplazamiento explica por qué el output no crece al ritmo esperado aunque la oferta externa parezca abundante.

El **Orchestration Load** debe descomponerse. ¿Se consume en especificar porque el outcome está mal definido? ¿En integrar porque los sistemas son opacos? ¿En revisar porque assurance no está diseñado? ¿En decidir porque los derechos regresan al founder? Cada causa exige una intervención distinta. Contratar otro project manager puede añadir una capa sin resolverla.

También conviene medir la cola. Un promedio moderado puede ocultar picos que paralizan a líderes durante lanzamientos o incidentes. Si todas las excepciones requieren la misma persona, el sistema no ha construido orquestación; ha ampliado su radio de dependencia.

La solución puede ser simplificar el portfolio, mejorar interfaces, elevar compromiso, automatizar checks, distribuir criterio o internalizar una capacidad. El objetivo no es orquestrar cualquier complejidad. A veces la mejor rutina es reducir el número de piezas.

Aprendizaje que abarata el siguiente ciclo

Una señal de que la orquestación se ha vuelto capacidad es que cada ciclo reduce ciertas fricciones del siguiente. La empresa formula briefs mejores, selecciona con más precisión, concede accesos de manera segura, detecta errores antes y transfiere contexto con menos esfuerzo. Si el trabajo de integración se repite igual, la experiencia no está acumulándose.

El aprendizaje puede residir en relaciones committed, no solo en documentos. Una red de confianza conoce estándares y plantea

objeciones antes. Pero la organización necesita suficiente memoria para que esa relación no sea insustituible por accidente. El equilibrio consiste en aprovechar contexto relacional y convertir las lecciones generalizables en infraestructura.

Tres medidas muestran la curva: *time-to-ramp* de nuevas fuentes comparables, número de aclaraciones repetidas y porcentaje de mejoras incorporadas a interfaz o estándar. No son benchmarks universales. Permiten observar si la organización está pagando menos impuesto de integración por unidad de outcome.

Assurance como diseño de información

Revisar todo no escala. No revisar nada no gobierna. Assurance debe colocar señales donde permiten distinguir output aceptable, excepción y daño severo. Puede combinar tests automáticos, muestreo, separación de funciones, observabilidad y revisión experta. La intensidad depende de reversibilidad y riesgo.

Con trabajo humano, la revisión debe respetar autonomía y evitar convertir feedback en vigilancia total. Con proveedores, necesita independencia suficiente para no delegar la definición de calidad. Con IA, requiere versionar modelo, datos, permisos y tasa de override. La forma cambia; la pregunta es la misma: ¿qué evidencia necesita el owner para aceptar o detener?

Una organización que no diseña assurance termina con dos extremos: confianza personal en fuentes conocidas y burocracia pesada para las nuevas. La primera concentra; la segunda bloquea acceso. Criterios explícitos permiten que la confianza se apoye en evidencia y que la apertura sea proporcional.

Reconfiguration como disciplina de salida

El loop suele celebrarse por su capacidad de añadir fuentes. Su prueba más difícil es retirarlas. Terminar un proveedor, desactivar una herramienta, reducir permisos o internalizar una pieza exige reconocer sunk costs y gestionar transición. La inercia mantiene configuraciones que dejaron de encajar.

Una revisión de reconfiguration debería incluir: trigger observado, performance frente al contrafactual, conocimiento que debe preservarse, consecuencias bilaterales, plan de continuidad y fecha de cierre. Si cambia la modalidad, registra por qué. Así la salida se convierte en evidencia para el siguiente *Source*, no en una ruptura olvidada.

La organización capaz de terminar bien mantiene mejores relaciones y más opciones. Orquestrar incluye cerrar el círculo.

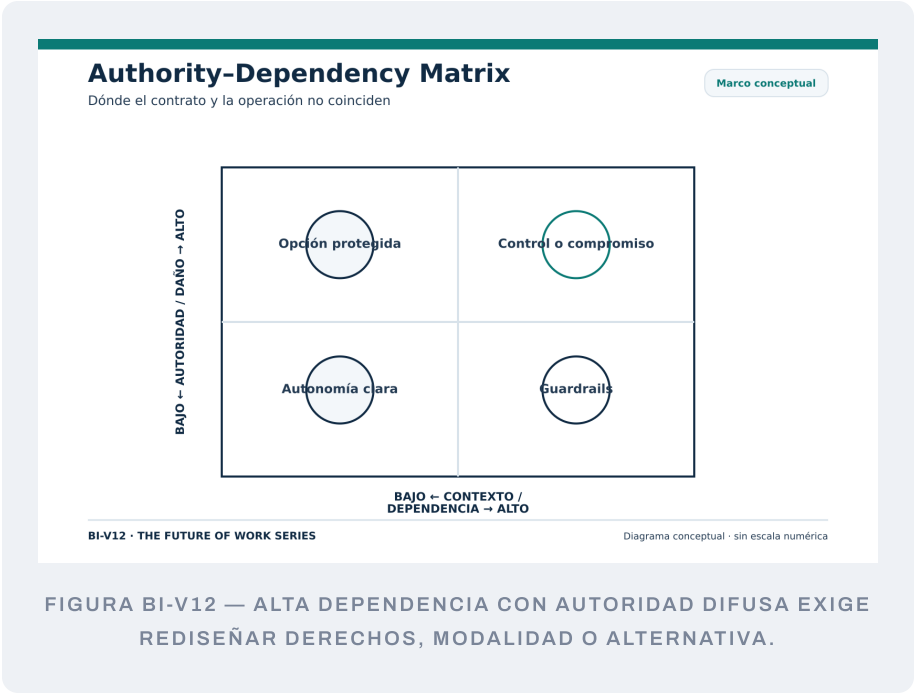
El acceso abre una posibilidad. Orquestrar significa convertirla en outcome, memoria y capacidad de cambiar, contando también el trabajo del integrador.

NOTAS DEL CAPÍTULO

1. Li & Fung, **Annual Reports**. Los informes corporativos documentan estrategia y reestructuración; el mecanismo causal exige fuentes independientes adicionales.
2. Robert Grant, “**Toward a Knowledge-Based Theory of the Firm**”, *Strategic Management Journal*, 1996.
3. David Sirmon et al., **resource orchestration research**.
4. David Teece, Gary Pisano y Amy Shuen, “**Dynamic Capabilities and Strategic Management**”, 1997.

Capítulo 12

LEADING WITHOUT EMPLOYMENT AUTHORITY



Relación, control y responsabilidad

Un crew cinematográfico puede reunir especialistas que no forman una plantilla permanente ni han trabajado juntos antes. Aun así, el rodaje no comienza como un grupo de desconocidos improvisando. Roles profesionales, reputación, lenguaje, secuencias de trabajo y expectativas crean una infraestructura social portátil. La investigación de Beth Bechky muestra cómo esas organizaciones

temporales coordinan mediante roles y prácticas compartidas.¹

El caso no convierte una producción en modelo universal. Enseña que la coordinación sin jerarquía común no ocurre sin institución. Parte de la estructura vive en ocupaciones, redes y rutinas anteriores al proyecto.

Cuando contributors no comparten relación laboral, el manager dispone de menos autoridad formal y puede existir más dependencia económica de la que el contrato admite. Liderar exige distinguir cuatro cosas: **derecho a dirigir, influencia, dependencia y accountability.**

Autoridad no es liderazgo

La autoridad contractual permite asignar o exigir ciertas conductas dentro de un marco. El liderazgo consigue que personas comprendan un outcome, confíen en reglas y coordinen decisiones. Se solapan, pero no son iguales.

Una empresa puede tener autoridad sobre un empleado y liderar mal. Puede carecer de autoridad laboral sobre un partner y coordinar de manera excelente mediante legitimidad, claridad y reciprocidad. También puede ejercer control de facto sobre una persona llamada independiente y crear riesgo jurídico y ético.

La primera regla es no usar cultura o colaboración para conseguir obediencia que el contrato no reconoce.

Authority–Dependency Matrix

El mapa utiliza dos ejes: grado de autoridad contractual/operativa y grado de dependencia bilateral.

CUADRANTE	RIESGO PRINCIPAL	PRÁCTICA NECESARIA
Autoridad alta / dependencia alta	captura, subordinación, daño de salida	derechos, protección, voz y revisión jurídica
Autoridad alta / dependencia baja	control excesivo para tarea modular	reducir prescripción, clarificar outcome
Autoridad baja / dependencia alta	accountability sin poder, proveedor crítico	reserva, governance, step-in, redundancia
Autoridad baja / dependencia baja	relación spot y variación	brief verificable, pago, disputa y salida

La dependencia se mide en ambos sentidos. La empresa puede depender de un especialista único; la persona puede depender de un solo cliente. Un contrato formalmente simétrico puede ocultar una economía muy asimétrica.

Contexto y legitimidad

Una persona externa acepta prioridades cuando entiende el propósito, considera legítimo el proceso y confía en que los acuerdos se respetarán. La empresa no puede exigir la misma identificación que a una plantilla; sí puede ofrecer contexto suficiente para decidir bien.

Contexto útil incluye:

- outcome y por qué importa;
- restricciones reales;
- decisiones ya tomadas y razones;
- stakeholders afectados;
- qué puede desafiarse;
- quién decide en conflicto;
- qué no se espera de la relación.

El último punto protege contra scope creep. Liderazgo relacional no significa disponibilidad moral ilimitada.

Incentivos más allá del precio

Pago importa y debe ser claro. También importan autonomía, previsibilidad, aprendizaje, reputación, reconocimiento, acceso futuro y calidad de la colaboración. Las partes pueden valorar cosas distintas.

Un incentivo eficaz no necesita manipular identidad. Puede ser un retainer que remunera reserva, un bonus ligado a outcome que la persona realmente controla, una renovación transparente o acceso a información que permite trabajar mejor. Los incentivos fallan cuando castigan variables fuera del control o crean competencia entre contributors interdependientes.

La medición debe reconocer el sistema. Si una agencia depende de aprobaciones lentas, medir solo su velocidad produce una ficción de accountability.

Cadencia y voz

Las personas externas suelen recibir información tarde: después de definir prioridad, cuando solo queda ejecutar. La organización pierde expertise y después se queja de que el partner “no piensa estratégicamente”.

La cadencia debe incluir a quien ejecuta en las decisiones que afectan a su output, sin convertir la relación en empleo de facto. La diferencia está en outcome y autonomía:

- compartir restricciones no es prescribir cada paso;
- invitar a challenge no es exigir pertenencia;
- acordar cadencia no es controlar toda la jornada;
- pedir trazabilidad no es vigilar continuamente.

Voz significa mecanismo para señalar riesgo, impugnar una evaluación o decir que el brief es imposible. En una red, la información incómoda puede ser más valiosa que la obediencia.

Confianza verificable

La confianza no es ausencia de control. Es expectativa basada en promesas cumplidas, competencia observada y recovery honesto. Una relación madura combina:

- criterios de aceptación;
- visibilidad proporcional al riesgo;
- autonomía en método;
- incidentes declarados;
- feedback bilateral;
- corrección y aprendizaje;
- consecuencias previsibles.

La vigilancia total puede reducir confianza y crear problemas de estatus. La opacidad total puede hacer imposible accountability. El diseño correcto observa el outcome y las señales necesarias, no cada movimiento.

Conflicto, apelación y salida

Una arquitectura externa necesita una teoría de conflicto. ¿Quién interpreta scope? ¿Cómo se disputa calidad? ¿Qué ocurre si el algoritmo de una plataforma limita acceso? ¿Existe apelación humana? ¿Quién paga trabajo aceptado parcialmente?

Platform governance demuestra que el intermediario puede asignar, monitorizar y evaluar trabajo mediante algoritmos. Eurofound y OIT describen cómo la gestión algorítmica automatiza funciones directivas y puede reducir contacto humano.² La eficiencia de matching no elimina la necesidad de explicación y recurso.

Para relaciones directas, el **Relational Leadership Contract** —no un contrato jurídico— registra:

- outcome y autonomía;
- información y cadencia;
- decision rights;
- feedback y evaluación;
- conflicto y apelación;
- pago y cambios de scope;
- salida, revocación y transferencia.

Cultura sin captura

Las empresas hablan de extender cultura a partners. Puede significar estándares compartidos, lenguaje, calidad y modo de resolver conflictos. También puede exigir lealtad, disponibilidad y trabajo emocional sin ofrecer derechos o seguridad equivalentes.

Una cultura permeable distingue **participación** de **pertenencia contractual**. No oculta diferencias. Un contributor puede comprender valores de producto y mantener independencia. Un proveedor puede desafiar una práctica. La institución es responsable de no convertir cultura en mecanismo de control informal.

La cultura más creíble con externos se observa en conducta:

- paga en plazo;
- comparte contexto relevante;
- respeta límites;
- admite errores;
- permite voz;
- asigna mérito;
- cierra relaciones dignamente.

No necesita un manifiesto para existir.

Control de facto

Las reglas de clasificación varían. En Estados Unidos, el IRS examina control conductual, control financiero y tipo de relación.³ La Recomendación 198 de la OIT enfatiza hechos de la relación, no solo forma contractual.⁴ La Directiva UE 2024/2831 aborda situación laboral y gestión algorítmica en platform work.⁵

Este libro no ofrece una regla jurídica global. Ofrece una alerta de diseño: si la empresa necesita prescribir cómo, cuándo y dónde trabaja una persona, exige exclusividad o integra la función en su actividad continua, debe revisar si la modalidad elegida corresponde a la sustancia. El framework no es una herramienta para evitar obligaciones.

Remote no equivale a external

GitLab y Automattic aparecen con frecuencia en conversaciones sobre nuevas workforces. Son útiles para estudiar documentación, distribución y trabajo asíncrono. También son un recordatorio: una plantilla remota sigue siendo empleo. Distribución espacial, relación

contractual y apertura de frontera son dimensiones distintas.

Una empresa puede ser remote-first y cerrada a capacidad externa. Otra puede compartir oficina y operar una red extensa. Mezclar ambas impide aprender qué mecanismo produce qué outcome.

Salida digna y memoria

La salida forma parte del liderazgo. Incluye:

- aviso y pago;
- devolución o exportación de artefactos;
- revocación de permisos;
- transferencia de decisiones y excepciones;
- feedback bilateral;
- comunicación a stakeholders;
- posibilidad de referencia o continuidad futura cuando corresponda.

Una salida punitiva destruye reputación en la red. Una salida sin memoria obliga a recomprar contexto. Una salida sin revocar accesos crea riesgo. El final de la relación es un test del sistema.

La objeción decisiva

Puede afirmarse que “liderazgo relacional” suaviza una realidad de poder. Una empresa grande sigue controlando presupuesto y acceso; hablar de confianza no cambia la dependencia. La objeción es correcta cuando el marco sustituye derechos por buenas intenciones.

Por eso la respuesta no es liderar mejor en abstracto. Es alinear conducta, contrato, pago, voz, protección y opción de salida. Allí donde la dependencia y el control sean altos, elevar garantías o cambiar modalidad.

La orquestación necesita relaciones legítimas. También necesita una empresa conectable. Ni la confianza ni un SLA compensan interfaces opacas, permisos improvisados o conocimiento inaccesible. La última parte del libro pregunta cómo diseñar una organización que pueda abrir capacidades sin abrir todo.

Liderar entre instituciones

Cuando una persona no trabaja para la empresa, no llega sin contexto institucional. Puede pertenecer a otra firma, a una profesión, a una comunidad o a un portfolio propio. Cada espacio tiene normas, incentivos y reputación. Liderar la colaboración significa conectar instituciones, no absorber a la contraparte en la cultura del comprador.

Un abogado externo conserva deberes profesionales; una comunidad abierta decide mediante su governance; una agencia responde a varios clientes; un independiente protege autonomía y reputación. La empresa puede definir outcome y restricciones, pero no asumir que su jerarquía se extiende intacta. Comprender el otro sistema ayuda a diseñar acuerdos realistas y a reconocer cuándo los objetivos no pueden alinearse.

La pregunta práctica es: ¿qué autoridad necesita el outcome y de dónde procede legítimamente? Puede proceder de contrato, expertise, reputación, estándar profesional o consenso. Si la única respuesta es poder económico, la relación será frágil aunque cumpla a corto plazo.

Un contrato psicológico explícito

Toda colaboración desarrolla expectativas que no aparecen en el contrato jurídico: rapidez de respuesta, disponibilidad, franqueza, reconocimiento, continuidad o discreción. Cuando quedan implícitas, cada parte interpreta el silencio a su favor. La empresa espera iniciativa; el profesional espera un scope estable. El partner espera renovación; el comprador cree que puede cambiar prioridad sin coste.

Un **working agreement** breve puede hacer visibles estas expectativas. Declara cadencia, tiempos de respuesta, canales, cambios, feedback, decisiones y límites. No sustituye asesoramiento legal ni crea una falsa relación laboral. Reduce la distancia entre el documento formal y el modo real de trabajar.

La revisión bilateral debe permitir renegociar. Una relación puede comenzar modular y volverse más interdependiente. Puede aumentar la reserva o aparecer dependencia. Si las expectativas cambian sin cambiar términos, el compromiso se financia de manera informal.

Poder para decir no

La calidad de una red depende de que la información contraria pueda circular. Un especialista que teme perder al cliente puede ocultar que el plazo es imposible. Un proveedor crítico puede evitar comunicar un incidente. Un sistema de reputación puede castigar la apelación. La autoridad económica distorsiona la señal que el leader necesita.

Diseñar voz significa crear un canal, proteger su uso y responder de forma previsible. Puede ser un derecho a escalar fuera del manager, una revisión de scope, una explicación de evaluación o una

parada por seguridad. No todas las relaciones necesitan estructuras formales, pero cuanto mayor sea la dependencia y el daño, mayor debe ser la protección.

La capacidad de decir no también mejora la especificación. Un partner con legitimidad para desafiar puede detectar un outcome incoherente antes de ejecutar. La obediencia rápida es a veces la forma más cara de coordinación.

El control algorítmico no cambia la sustancia

Software puede asignar trabajo, medir tiempo, ordenar personas y bloquear acceso. La intermediación técnica no elimina poder directivo. En algunas plataformas, la gestión algorítmica crea precisamente la necesidad de transparencia, contacto humano y recurso que refleja la regulación europea de trabajo en plataformas.

Dentro de una empresa, automatizar evaluación de proveedores o contributors requiere la misma cautela. Una puntuación puede combinar señales incompletas, reforzar sesgos y negar contexto. Si afecta a acceso económico o reputación, debe existir explicación, revisión y corrección proporcionadas.

El principio no es prohibir métricas. Es evitar que una interfaz tecnológica transforme una decisión disputable en hecho aparentemente objetivo. La accountability permanece en quienes diseñan y utilizan el sistema.

Cultura como protocolo de reciprocidad

Extender cultura puede significar pedir que todos utilicen los mismos slogans. Es más útil definir conductas recíprocas: cómo se comparte contexto, cómo se reconoce autoría, cómo se paga, cómo

se declara un error y cómo se resuelve desacuerdo. Estas prácticas son observables y no exigen fingir una pertenencia contractual común.

La reciprocidad importa porque la empresa también es evaluada. Una red de especialistas decide si volver, recomendar o reservar capacidad. Pagar puntualmente y cerrar bien no son cortesía adicional; sostienen el committed network que reduce latencia y riesgo.

Una cultura permeable admite identidades distintas y conserva estándares comunes sobre el outcome. Ese equilibrio produce cooperación sin apropiarse de la persona.

La transición de manager a integrador

El manager de empleados conserva responsabilidades que no desaparecen: desarrollar, priorizar, cuidar condiciones y ejercer autoridad legítima. Cuando coordina una capability plural, añade otra función: diseñar interfaces entre relaciones distintas. No se convierte universalmente en “orchestrator” ni deja de gestionar personas.

Debe saber cuándo usar cada lógica. Con empleados, puede reasignar dentro del marco del puesto y tiene deberes de desarrollo. Con un proveedor, negocia outcome y governance. Con un independiente, respeta autonomía compatible con la modalidad. Con IA, define permisos y supervisión. Aplicar la misma conducta a todos crea incoherencia jurídica y humana.

El liderazgo maduro no reduce las modalidades a recursos equivalentes. Mantiene sus diferencias y las conecta alrededor de una promesa común.

Sin empleo común, la coordinación depende más de legitimidad, contexto, voz y acuerdos; la etiqueta contractual nunca borra el control real ni la responsabilidad.

NOTAS DEL CAPÍTULO

1. Beth Bechky, “Gaffers, Gofers, and Grips”, *Organization Science*.
2. OIT, *Algorithmic management in the workplace*; Eurofound, *Platform work: Algorithmic management*.
3. Internal Revenue Service, *Employee or independent contractor?*.
4. OIT, *Recomendación 198 sobre la relación de trabajo*.
5. Unión Europea, *Directiva (UE) 2024/2831*.

HERRAMIENTA — ORCHESTRATION CAPABILITY REVIEW

Elija una capacidad cuyo outcome haya fallado o se haya retrasado.

1. Recorra Sense, Specify, Source, Integrate, Assure, Learn y Reconfigure.
2. Identifique output ausente, failure mode y evidencia en cada rutina.
3. Registre horas del núcleo dedicadas a orquestar.
4. Clasifique capability debt: knowledge, interface, assurance, concentration, accountability u orchestration.
5. Complete Authority-Dependency Matrix para cada relación material.
6. Defina una acción, owner, medida y fecha; no produzca una puntuación agregada.

El techo de la orquestación

Una empresa pequeña amplía su red. Añade una agencia, varios especialistas, herramientas de IA y proveedores funcionales. El founder siente que puede movilizar una organización mucho mayor. Durante unas semanas, el leverage es real. Después el calendario se llena de briefs, revisiones, decisiones y reuniones. El cuello de botella ya no es ejecución. Es contexto.

Este techo aparece cuando el coste marginal de integrar cada fuente crece casi tan deprisa como la capacidad añadida. La organización tiene más manos y la misma cabeza.

Contar la carga invisible

Durante un mes, el núcleo registra tiempo de Sense, Specify, Source, Integrate, Assure, Learn y Reconfigure. No para facturar internamente cada minuto, sino para localizar el impuesto. Puede descubrir que selecciona poco y reexplica mucho; que los outputs llegan rápido y esperan aceptación; que la automatización genera más casos para revisar; o que ningún aprendizaje reduce el siguiente ciclo.

La carga no es mala por definición. Expertise externo necesita integración. El problema es que permanezca invisible y se atribuya al modelo una escala que depende de trabajo senior no sostenible.

El registro debe incluir picos y tiempo de recuperación. Una media semanal oculta lanzamientos que monopolizan al founder.

Cuatro respuestas equivocadas

La primera es añadir más fuentes. Si el bottleneck es decisión, aumenta la cola. La segunda es añadir un coordinador sin derechos ni contexto; crea otra interfaz. La tercera es documentar todo de golpe; traslada la carga a producción de manuales. La cuarta es exigir autonomía a las fuentes sin dar outcome, permisos o authority.

La respuesta depende del mecanismo. Ambigüedad se reduce con prioridades y briefs. Handoffs se reducen rebundleando. Review se escala con assurance basado en riesgo. Decisiones se distribuyen con rights y criterios. Contexto recurrente se convierte en interfaz y memoria. Complejidad sin valor se elimina.

La orquestación no justifica una red arbitrariamente grande. Incluye la disciplina de decir que una capacidad debería estar más cerca o que dos piezas deben reunirse.

Construir integradores

Un founder no puede ser el único traductor. La empresa necesita personas y relaciones capaces de poseer outcomes end-to-end. Un integrador combina suficiente dominio, autoridad y visibilidad para resolver interfaces. No es un project manager que solo persigue fechas.

La capacidad puede distribuirse mediante células, owners o partners committed. Necesita redundancia: si un integrador falta, la organización no debería perder todo el mapa. Decision logs, pairing y rotación ayudan, pero también una estructura que limita el número de capacidades por owner.

Desarrollar integradores exige exposición a excepciones y

trade-offs. Automatizar todo lo rutinario sin rediseñar aprendizaje puede crear escasez de quienes saben juzgar.

Confianza y assurance

Cuando la carga crece, la empresa puede confiar más en fuentes conocidas. Esa estrategia reduce revisión y concentra dependencia. O puede aplicar controles uniformes a todas, elevando burocracia. La solución es assurance proporcional y basado en evidencia.

Una relación con historial puede recibir más autonomía dentro de límites; un output de bajo daño puede muestrearse; una decisión crítica mantiene doble revisión. La confianza se gana y se observa. No sustituye logs, salida o accountability.

El objetivo es liberar atención humana para excepciones. Si el control consume más capacidad que la ejecución, el sistema necesita otro diseño.

Liderazgo sin obediencia total

El founder bajo presión puede empezar a prescribir cada paso a fuentes externas. La autoridad informal crece y la modalidad deja de encajar. O puede abdicar y esperar iniciativa estratégica de personas contratadas para un módulo. Ambos extremos nacen de interfaces y expectativas débiles.

El Relational Leadership Contract clarifica outcome, autonomía, cadencia, voz y escalado. La empresa paga la reserva que necesita y no convierte cultura en disponibilidad ilimitada. La contraparte conoce quién decide y puede señalar que el sistema es imposible.

La calidad de la relación reduce Orchestration Load porque la información viaja antes y el conflicto tiene ruta. La buena voluntad

sin derechos no basta.

Romper el techo

El techo se rompe cuando la organización aprende a producir capacidad de segundo orden: interfaces reutilizables, owners distribuidos, committed networks, assurance escalable y memoria. El coste marginal de la siguiente conexión disminuye. No llega a cero y puede volver a crecer con complejidad.

Una empresa debe observar si el leverage aumenta o solo se desplaza. La señal no es cuántas fuentes coordina una persona, sino cuántos outcomes puede gobernar el sistema sin degradar calidad, aprendizaje, condiciones o recuperación.

Si la respuesta exige una estructura más estable, no es una derrota del modelo. Es la evidencia que el modelo debía producir.

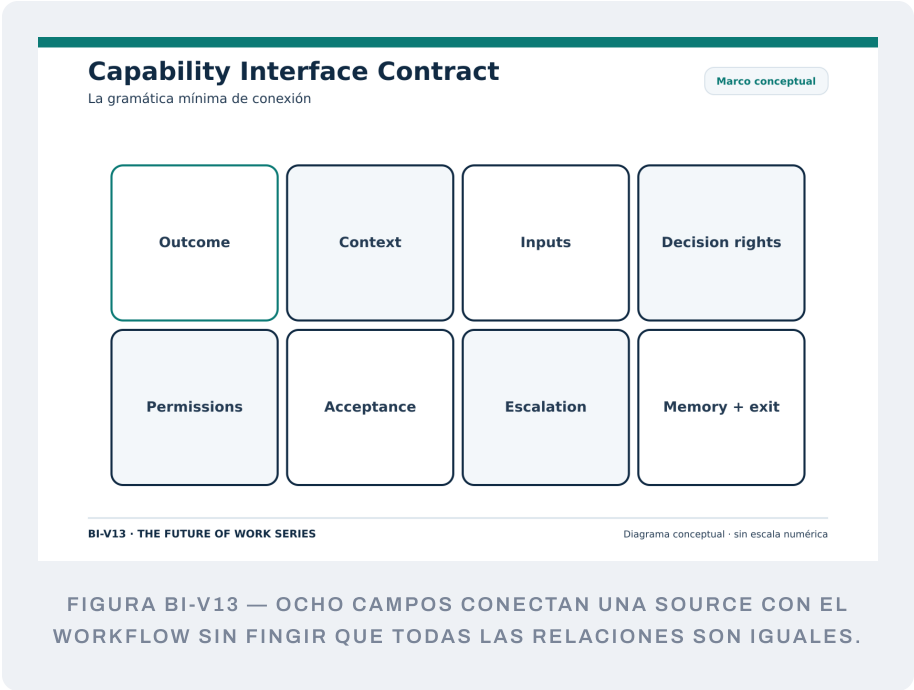
PARTE V

LA ORGANIZACIÓN PERMEABLE

Una organización conectable no es una organización abierta por defecto. Es capaz de abrir una capacidad con interfaces, permisos, assurance y salida, y de cerrar otra cuando el riesgo lo exige.

Capítulo 13

DESIGNING FOR ACCESS



The Capability Interface Contract

Una contribución externa puede pasar más tiempo esperando que ejecutando: esperando acceso, una decisión, datos, respuesta a una duda o confirmación de que el output sirve. Un empleado puede sufrir exactamente lo mismo. Estar dentro de la frontera contractual no garantiza una buena interfaz; estar fuera vuelve su ausencia más visible.

Las organizaciones suelen transferir al contributor el trabajo de

descubrir cómo funcionan. “Pregunta a alguien”, “mira el canal”, “usa el ejemplo anterior”. Quien ya pertenece a la empresa sobrevive gracias a contexto informal. Quien llega desde fuera consume su capacidad en arqueología.

Diseñar para access no significa documentarlo todo. Significa estabilizar lo suficiente para que una fuente pueda llegar a ser productiva, segura y recuperable. El artefacto central es el **Capability Interface Contract**. No es necesariamente un contrato jurídico. Puede componerse de brief, runbook, schema, policy, sandbox, checklist, SLA y conversaciones.

Modularidad sin fantasía

Carliss Baldwin y Kim Clark explican cómo modularidad y design rules permiten cambiar componentes dentro de una arquitectura.¹ En organización, la metáfora es útil con un límite. Las personas y los problemas ambiguos no son piezas de software. Una interfaz puede reducir dependencias evitables; no vuelve codificable todo el conocimiento ni elimina relación.

Antes de diseñarla, pregunte:

- ¿qué dependencias pueden estabilizarse?;
- ¿qué conversaciones deben permanecer?;
- ¿qué incertidumbre es productiva?;
- ¿qué ambigüedad solo refleja descuido?;

Una interface excelente para traducción repetitiva puede ser insuficiente para estrategia. En trabajo exploratorio, la interfaz puede ser un equipo comprometido y un ritmo de aprendizaje, no una especificación cerrada.

Los ocho campos

1. OUTCOME

Define estado deseado, usuario, plazo y criterios de aceptación. No “entregar informe”, sino qué decisión o cambio permitirá el informe. También declara errores severos y externalidades.

2. CONTEXT

Incluye historia relevante, decisiones previas, restricciones, unknowns y vocabulario. Contexto no es un volcado de documentos. Es lo que cambia la decisión.

3. INPUTS AND DEPENDENCIES

Datos, sistemas, handoffs, secuencia, formatos y owners. Indique calidad conocida y qué hacer cuando falta un input. Una interfaz que presupone datos limpios puede desplazar el fallo.

4. DECISION RIGHTS

Qué puede decidir la fuente, qué propone, qué debe escalar y quién responde. Incluye tiempos de decisión; un owner que tarda dos semanas puede ser el verdadero cuello de botella.

5. PERMISSIONS

Acceso mínimo necesario, datos, IP, herramientas, duración, audit trail, revocación y entorno seguro. *Least privilege* no significa impedir trabajar; significa conceder lo suficiente con alcance y caducidad claros.

NIST advierte que riesgo de supply chain digital debe gestionarse a través de productos y servicios de terceros.² La interfaz de capacidad forma parte de ese control: identidad, dependencia y

salida importan tanto como el contrato comercial.

6. ACCEPTANCE AND OBSERVABILITY

Cómo se evalúa, quién revisa, qué logs existen, qué SLA importa y cómo se declara incidente. El estándar debe distinguir defectos comunes y errores severos. Observability no es vigilancia de la persona; es visibilidad del sistema y del outcome.

7. FEEDBACK AND ESCALATION

Cadencia, canal, autoridad y tiempo de respuesta. Define qué ocurre cuando scope cambia o una condición hace imposible el estándar.

8. MEMORY AND EXIT

Qué artefactos y decisiones regresan, dónde quedan, cómo se transfieren, qué acceso se revoca y cuál es el recovery plan. La salida se diseña cuando entra la capacidad, no después de una ruptura.

Dos carriles

El Capability Interface Contract tiene un carril de **execution** y otro de **assurance**.

Execution conecta inputs, método permitido, handoffs y output. Assurance conecta aceptación, reviewer, seguridad, incidentes y accountability. Mezclarlos crea conflicto: la misma fuente produce y se certifica.

En trabajo de bajo daño, assurance puede ser una prueba automática. En un proceso regulado, necesita revisión cualificada y trazabilidad. La interfaz escala con el riesgo.

GitLab y el coste de ser visible

El handbook público de GitLab ilustra documentación como infraestructura. Una organización all-remote no puede depender igual de conversaciones locales y memoria oral; necesita hacer visibles prácticas, responsabilidades y decisiones.³

La evidencia corporativa describe la intención y los artefactos. No prueba que cada página se use ni que documentación cause performance. Mantenerla cuesta. Puede crecer más rápido que la capacidad de leer, contener contradicciones o convertir una excepción pasada en regla permanente.

La lección no es “escriba un handbook enorme”. Es medir la interfaz por conducta:

- ¿una persona nueva encuentra el owner?;
- ¿puede actuar sin pedir contexto repetido?;
- ¿las reglas coinciden con práctica?;
- ¿un cambio actualiza el artefacto?;
- ¿la salida deja memoria recuperable?

La documentación que no reduce una fricción observada es candidata a eliminación.

NASA y la especificación incompleta por diseño

Los challenges de NASA muestran una interfaz diferente: problema, criterios, reglas y mecanismo de evaluación permiten participar a una red. La interfaz no contiene toda la misión ni concede acceso a cada sistema. Delimita una unidad.

Después del resultado, la integración continúa. Si el challenge entregó una solución, la institución todavía debe decidir cómo

verificar, adaptar y operar. La interfaz de sourcing no equivale a interface de producción.

Ese punto es importante con concursos, agencias y IA. El prompt o brief inicial puede producir output; el Capability Interface Contract abarca su vida dentro del workflow.

Trabajo tácito y regulado

En salud, manufactura o servicios profesionales regulados, una interfaz escrita puede ser necesaria e insuficiente. El juicio se aprende mediante práctica, supervisión y exposición a excepciones. El entorno físico, el paciente, la máquina o la responsabilidad legal no se reducen a un schema.

Diseñar para access puede concluir:

- pairing obligatorio;
- periodo de shadowing;
- certificación;
- supervisión presencial;
- relación committed;
- prohibición de separar ciertas decisiones;
- control estable.

El framework no existe para abrir todo. Existe para descubrir cuándo abrir requiere más institución de la que la opción aporta.

Interface debt

Una interfaz débil crea deuda:

- **ambiguity debt:** briefs distintos para la misma capacidad;
- **permission debt:** accesos permanentes o retrasos recurrentes;
- **acceptance debt:** calidad discutida después de entregar;
- **memory debt:** decisiones perdidas;
- **exit debt:** datos y artefactos no exportables;
- **human debt:** contributors absorben el coste de interpretar caos.

Registre incidentes y horas. La deuda se vuelve priorizable cuando muestra efecto en ramp, defectos o recovery.

Validar la interfaz

El claim central es una hipótesis moderadora: con igual complejidad y calidad de fuente, una interfaz suficiente debería reducir ambigüedad evitable y acelerar ramp. Para probarla, compare cambios en una misma capacidad:

- tiempo hasta primer outcome aceptado;
- preguntas repetidas;
- permisos excepcionales;
- first-pass acceptance;
- defectos y severidad;
- horas de manager;
- time-to-recovery;
- conocimiento recuperable al salir.

Una mejora puede deberse a una fuente mejor o a un problema más simple. El test no resolverá causalidad perfecta, pero evita declarar

éxito por completar un template.

La objeción de la burocracia

Una interface completa puede convertirse en barrera de acceso. Cada campo añade trabajo, y el contributor puede terminar satisfaciendo el proceso en lugar del cliente. Además, documentar puede crear falsa confianza: el mundo real cambia.

La respuesta es un contrato vivo y proporcional. Empiece con lo mínimo que permite actuar con seguridad. Añada un campo cuando un fallo lo justifique. Elimine reglas sin función. Mantenga una conversación para aquello que no debe congelarse.

La mejor interfaz no es la más exhaustiva. Es la que hace explícitas las dependencias decisivas y facilita adaptación.

Una interfaz permite una conexión. Una empresa puede tener diez interfaces excelentes y seguir abriendo cada capacidad mediante favores, excepciones y memoria personal. La siguiente pregunta cambia de unidad: **¿qué hace que la organización entera sea permeable sin convertir “más abierta” en objetivo?**

Una interfaz mínima viable

Los ocho campos no deben convertirse en ocho documentos. Para una capacidad reversible y de bajo riesgo, una interfaz mínima puede caber en una página: outcome, owner, inputs, acceptance, permisos y salida. Para una capacidad crítica, cada campo se desarrolla con tests, runbooks, rights y responsables. La diferencia es materialidad, no preferencia por la documentación.

La interfaz mínima debe permitir que una persona nueva responda seis preguntas sin arqueología: qué debe lograr, qué

recibe, qué puede decidir, cómo se comprueba, a quién escala y qué deja al terminar. Si alguna requiere conocer personalmente al fundador, existe dependencia. Si todas requieren leer cien páginas, existe otra forma de dependencia.

El diseño comienza observando fricciones reales. Registre durante onboarding qué preguntas se repiten, qué permisos esperan, qué decisiones bloquean y qué errores nacen de ambigüedad. La interfaz se construye para reducir esas fricciones, no para anticipar cada contingencia. Cada campo debe tener un usuario y un fallo que evita.

Interfaces diferentes para relaciones diferentes

Un empleado nuevo, una firma profesional, una comunidad y un sistema de IA pueden necesitar outcome, contexto y feedback. No necesitan la misma interfaz institucional. El empleo incluye desarrollo, autoridad y deberes del empleador. Un proveedor necesita scope, governance y términos comerciales. Una comunidad conserva autonomía y reglas propias. Un sistema técnico necesita identidad, permisos, versionado, logs y fallback.

Utilizar una única plantilla puede borrar derechos. Llamar “API” a una persona sugiere invocación sin reciprocidad; tratar un agente como colega puede ocultar que no posee juicio moral ni responsabilidad. La interfaz comparte una gramática operacional y conserva campos específicos de modalidad.

El Capability Interface Contract debe, por tanto, tener un núcleo común y anexos. Núcleo: outcome, inputs, decisiones, acceptance, feedback, memoria y salida. Anexo humano: autonomía, disponibilidad, voz, crédito y condiciones. Anexo proveedor: SLA, subcontratación, auditoría y continuidad. Anexo IA:

modelo/versión, herramientas, límites, supervisión, errores severos y kill switch.

El presupuesto de contexto

Contexto no es gratuito ni infinito. Cada fuente necesita una parte para decidir bien; compartir demasiado aumenta tiempo, confidencialidad y riesgo. La empresa debe construir un **context budget**: información mínima para el outcome, con razones y caducidad.

El presupuesto distingue tres niveles. **Contexto de tarea**: inputs y restricciones inmediatas. **Contexto de workflow**: dependencias, excepciones y usuarios. **Contexto estratégico**: razones, trade-offs y decisiones futuras. Una tarea modular puede requerir el primero; un partner committed, los tres. El nivel debe corresponder a responsabilidad.

El criterio de minimización no consiste en ocultar. Consiste en compartir aquello que cambia la calidad y evitar volcar repositorios sin guía. Un brief breve con decisiones relevantes puede integrar mejor que acceso indiscriminado a miles de documentos.

Permisos como producto

El acceso técnico suele ser el tramo más lento y arriesgado de una colaboración. Cuentas prestadas, privilegios permanentes o aprobaciones manuales destruyen velocidad y control. Tratar permisos como producto significa disponer de roles predefinidos, entornos seguros, expiración, logging y un proceso de revocación.

La arquitectura debe permitir trabajar con *least privilege* sin bloquear. Si cada nuevo contributor necesita una excepción, el

modelo de permisos no refleja los workflows reales. Si todas las fuentes reciben acceso amplio porque es más sencillo, la empresa financia velocidad con superficie de riesgo.

La salida prueba la calidad. ¿Puede revocarse una identidad sin perder artefactos? ¿Sabemos qué datos se copiaron? ¿Permanecen automatizaciones o tokens? Un inventario unido al owner y al outcome reduce tanto la latencia de entrada como la incertidumbre de cierre.

El owner de la interfaz

Las interfaces se degradan cuando nadie las posee. Cambia el sistema, el documento sigue igual; aparece una excepción, se resuelve en privado; rota el partner, la memoria se pierde. El owner no tiene que escribir todo. Debe observar performance y decidir qué cambio merece convertirse en infraestructura.

Tres señales ayudan: *ramp time*, *first-pass acceptance* y *recovery time*. Si mejoran, la interfaz puede estar cumpliendo. Si no, se investiga qué parte falla. Una interfaz también puede ser demasiado rígida: alta conformidad y baja innovación. El feedback de contributors permite detectar cuándo las reglas dejaron de servir.

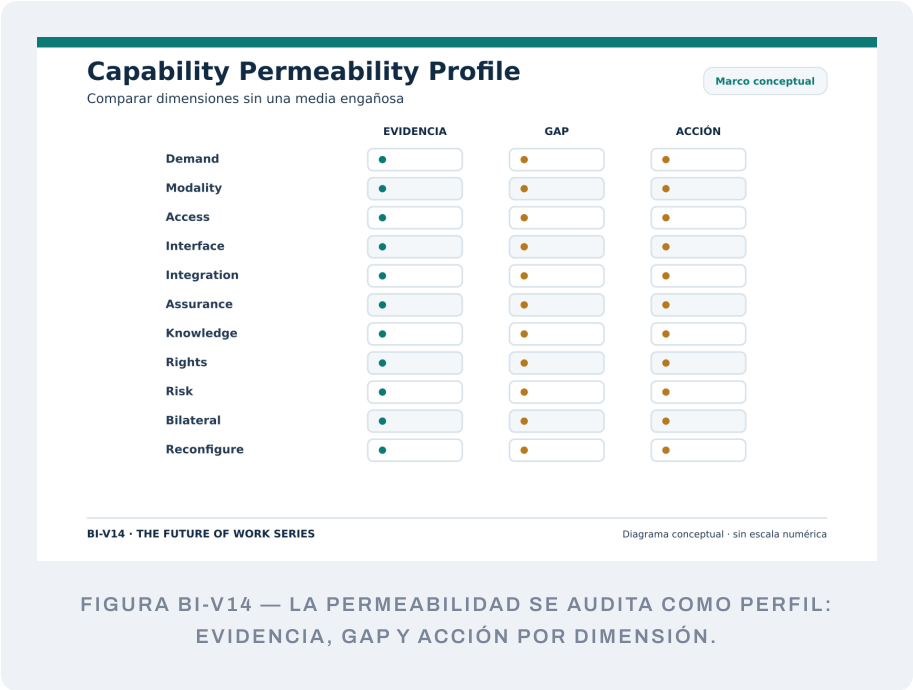
La interfaz es un activo compartido. Quien llega desde fuera suele ver contradicciones que los internos normalizaron. Incorporar esa perspectiva convierte onboarding en investigación organizativa.

Ser conectable cuesta. Una interfaz útil reduce ambigüedad, limita permisos, hace observable la calidad y conserva memoria; no convierte todo trabajo en módulo.

NOTAS DEL CAPÍTULO

1. Carliss Baldwin y Kim Clark, *Design Rules*, MIT Press.
2. NIST, SP 800-161 Rev. 1 — Cybersecurity Supply Chain Risk Management Practices.
3. GitLab, [About GitLab](#) y recursos sobre [remote work](#).

THE PERMEABLE COMPANY



Abrir y cerrar capacidades con intención

Dos empresas compran trabajo externo. En una, la relación entra por una recomendación personal; el contributor recibe accesos prestados, el scope vive en mensajes y la memoria queda en quien lo contrató. En la otra, un owner define la capacidad, existe una interfaz, assurance, permisos con caducidad y una salida. Ambas externalizan. Solo una está desarrollando permeabilidad gobernada.

La comparación es un arquetipo, no evidencia de superioridad.

Sirve para distinguir transacción y propiedad organizativa. Una **permeable company** puede decidir qué capacidad cruza la frontera, en qué dirección, mediante qué gate y con qué control. También puede cerrarla cuando el riesgo o el aprendizaje lo exige.

Por qué “open” es demasiado amplio

Open company puede significar transparencia, open source, participación, innovación abierta, datos abiertos o workforce externa. Existe además un **Open Organization Maturity Model** asociado al movimiento de open organizations.¹ Presentar el mismo nombre como concepto propio sería confuso.

Permeability señala una propiedad más estrecha: flujos de trabajo, conocimiento, datos, decisiones y capacidad a través de fronteras. Una empresa puede ser transparente y poco permeable en ejecución. Puede usar open source y cerrar datos. Puede abrir innovación y mantener producción bajo control.

La pregunta no es “¿cuán abierta es?”. Es “¿qué cruza, por qué, bajo qué evidencia y quién responde?”.

Cinco niveles, no una escalera moral

NIVEL 0 — PAYROLL DEFAULT

La organización parte de puestos. Trabajo externo aparece como excepción contable y no se conecta al planning de capacidades.

Valor: claridad y control en contextos apropiados.

Riesgo: talento y dependencia fuera de payroll permanecen invisibles.

NIVEL 1 — AD HOC ACCESS

Equipos acceden mediante relaciones personales, urgencias o shadow procurement. Puede ser rápido.

Valor: experimentación y respuesta local.

Riesgo: permisos, calidad, clasificación, conocimiento y continuidad improvisados.

NIVEL 2 — GOVERNED ACCESS

Existen criterios, contratos, onboarding, seguridad y owners. Procurement y HR reconocen modalidades distintas.

Valor: repetibilidad y control.

Riesgo: burocracia sin aprendizaje; gates diseñados por categoría y no por capacidad.

NIVEL 3 — ORCHESTRATED PORTFOLIO

La empresa decide por capability, ve composición y concentración, usa interfaces y assurance y registra Total Access Cost.

Valor: coherencia de sistema.

Riesgo: sobrecargar al núcleo orquestador o convertir frameworks en reporting.

NIVEL 4 — ADAPTIVE CAPABILITY SYSTEM

Gates, memoria, triggers, bilateral outcomes y reconfiguration operan con evidencia. La empresa puede abrir o cerrar selectivamente.

Valor: aprendizaje y adaptación.

Riesgo: inestabilidad, optimización excesiva o cambios que erosionan identidad.

Nivel 4 no es destino universal. Una capacidad regulada puede operar responsablemente en Nivel 2 con control estable. Una tarea reversible puede permanecer ad hoc. La madurez es fit, no apertura máxima.

Once dimensiones

La escalera resume. El instrumento real es un heatmap por capacidad y dimensión.

1. **Demand:** outcome y baseline antes de pedir recurso.
2. **Modality:** elección BBBA con alternativa y trigger.
3. **Access:** profundidad, latencia y condiciones bilaterales.
4. **Interface:** contexto, inputs, permisos, aceptación y salida.
5. **Integration:** time-to-ramp, handoffs y management load.
6. **Assurance:** calidad, seguridad, errores severos y fallback.
7. **Knowledge:** creación, retención y transferencia.
8. **Decision rights:** owner, reviewer, escalation y accountable entity.
9. **Risk:** concentración, clasificación, privacidad, continuidad y terceros.
10. **Bilateral outcomes:** valor para firma y efectos sobre contributors.
11. **Reconfiguration:** señal, decisión, transición y recovery.

No se calcula una media. Una nota total permitiría que buena documentación compensara ausencia de accountability. Cada dimensión tiene un mínimo según el riesgo.

GitLab, P&G y Red Hat: permeabilidades distintas

GitLab hace visibles prácticas de trabajo distribuido mediante documentación. Eso puede facilitar conexión y transferencia, pero no demuestra apertura de toda capacidad ni confunde remote employment con sourcing externo.

P&G abre flujos de innovación mediante Connect + Develop y conserva filtros, IP, adopción y capacidades internas. Su permeabilidad es selectiva: conocimiento e invenciones pueden entrar; decision rights y responsabilidad permanecen estructurados.

Red Hat y Linux muestran flujos bidireccionales entre empresa y comunidad, además de empleo, producto, soporte y governance. Producción abierta e institución comercial coexisten.

Los tres casos desmienten una única forma. Permeabilidad puede residir en documentación, innovación o producción comunitaria. Cada una necesita gates y mecanismos de captura/redistribución de valor distintos.

El fallo como dato

Una empresa no demuestra madurez con una encuesta de autopercepción. La demuestra en decisiones e incidentes:

- un acceso bloqueado por riesgo con rationale;
- un contributor incorporado con menor ramp;
- una dependencia descubierta y diversificada;
- conocimiento transferido durante salida;
- una capacidad internalizada tras fallar un gate;
- una automatización detenida por error severo;
- condiciones mejoradas al detectar riesgo bilateral.

El maturity review debe incluir dos evaluadores y evidencia. Las discrepancias revelan shadow systems: la policy dice Nivel 2, el equipo trabaja en Nivel 1.

El optimum contextual

Para cada capacidad defina un perfil objetivo, no el máximo. Ejemplo:

- datos sensibles: alto en permissions y assurance, acceso más restringido;
- diseño exploratorio: alto en demand y feedback, interface flexible;
- traducción repetitiva: alta modularidad y mercado abierto con QA;
- incident response: committed network, reserva y recovery probado;
- criterio estratégico: Solid Minimum y aprendizaje cercano.

El perfil cambia por país y fecha. Una nueva regulación puede elevar governance. Un mercado más profundo puede reducir latencia. Un

incidente puede revelar que el nivel objetivo era insuficiente.

Abrir y cerrar

La permeabilidad se mide también por la capacidad de cerrar sin caos. Revocar permisos, transferir memoria, cumplir pagos y preservar continuidad forman parte de la propiedad. Una puerta que solo abre no es una interfaz; es una vulnerabilidad.

Cerrar puede significar:

- internalizar una decisión;
- retirar una fuente;
- bloquear una herramienta;
- reducir datos compartidos;
- terminar una comunidad o partnership;
- elevar reserva o exclusividad;
- simplificar el workflow.

Una organización adaptable puede volverse menos abierta en una capacidad y más en otra. No hay contradicción si el criterio es outcome y responsabilidad.

La objeción contra los maturity models

Los maturity models convierten complejidad en escalera y venden el nivel superior. Las empresas aprenden a responder lo correcto sin cambiar conducta. Este riesgo no se corrige con más preguntas.

Por eso el **Capability Permeability Maturity Model** impone cuatro restricciones:

1. unidad = capacidad, no empresa abstracta;
2. evidencia = artefactos, decisiones, tiempos e incidentes;
3. output = gaps y acciones, no ranking;
4. optimum = contextual, no Nivel 4 universal.

Si el instrumento no cambia una decisión, no merece llamarse herramienta.

Permeabilidad y responsabilidad

La empresa permeable no es hueca. Mantiene un Solid Minimum, sabe qué cruza la frontera, observa performance y retiene memoria. Cuanto más puede movilizar, más importante es conocer el límite de lo que puede gobernar.

El libro comenzó con el headcount. Ahora vemos un sistema de capacidades. La nómina no desaparece; ocupa una capa dentro de una arquitectura que incluye relaciones comprometidas, acceso y automatización. Falta reunir el cambio y evitar su lectura más peligrosa: que *Beyond Payroll* sea un manual sofisticado de externalización.

Permeabilidad en cuatro direcciones

Abrir la empresa suele imaginarse como permitir que talento y conocimiento entren. La permeabilidad opera en cuatro direcciones.

Entrada: ideas, trabajo, tecnología y datos. **Salida:** conocimiento, pagos, estándares, reputación y contribuciones. **A través:** coordinación entre partes que no pasan por un centro único. **Cierre:** capacidad para detener el flujo, revocar y recuperar.

Una organización puede ser buena en entrada y mala en salida:

consume contribuciones sin devolver feedback ni valor. Puede abrir datos hacia un proveedor y no recuperar decisiones. Puede permitir colaboración transversal y carecer de autoridad para resolver conflictos. El heatmap debe observar la dirección, no solo la existencia del vínculo.

La reciprocidad importa en comunidades y networks. Si la empresa extrae valor sin participar en mantenimiento o governance, la fuente puede degradarse o resistir. La permeabilidad sostenible reconoce que otros actores también fijan reglas.

Selectividad explícita

Una compañía madura puede publicar código y cerrar datos; abrir challenges y mantener decisiones dentro; contratar producción y controlar calidad; usar IA para preparación y prohibir acciones autónomas. Estas combinaciones no son incoherentes. Expresan una política de selectividad.

La política puede formularse por objeto: trabajo, dato, conocimiento, decisión, identidad y capital. Para cada uno declara qué puede cruzar, bajo qué gate y qué nunca cruza sin una aprobación material. Así evita dos extremos: apertura ad hoc dirigida por urgencia y cierre general dirigido por miedo.

La selectividad necesita excepciones, porque ningún mapa anticipa todo. Pero una excepción debe tener owner, duración y revisión. Cuando las excepciones se repiten, el perfil objetivo o la interfaz están mal diseñados.

La permeabilidad cambia el poder

Abrir una frontera no solo mejora acceso; redistribuye capacidad de

decidir. Una plataforma puede ordenar participantes. Un partner que controla datos puede condicionar estrategia. Una comunidad puede rechazar el roadmap de la empresa. Un contributor puede llevar conocimiento valioso a otros clientes. Estos efectos son parte de la arquitectura, no ruido.

El review debe preguntar quién establece reglas, quién puede apelar, quién captura valor y quién absorbe el fallo. Una organización puede aumentar permeabilidad operativa y concentrar governance. Puede reducir control directo y ganar influencia mediante estándares o reputación. El resultado no se resume en “más abierta”.

Esta dimensión política también existe dentro. Equipos que antes controlaban recursos pueden perder exclusividad cuando otros acceden al mercado. Procurement, HR y tecnología pueden disputar ownership. La transición necesita una conversación sobre decisiones, no solo nuevas herramientas.

Cerrar sin castigar

Una empresa que puede cerrar bien conserva mejores opciones futuras. Terminar una colaboración con pago, explicación, transferencia y revocación protege continuidad y reputación. Cerrar de forma abrupta o punitiva enseña a la red a retener información y evitar dependencia.

El cierre puede obedecer a performance, cambio de estrategia, internalización, regulación o desaparición de la necesidad. Cada motivo exige un proceso distinto. Un incidente severo puede requerir bloqueo inmediato y revisión posterior; un cambio ordinario permite transición. Diseñar ambos evita improvisar bajo presión.

La capacidad de cerrar también protege a contributors. Deben conocer aviso, propiedad de trabajo, datos y pagos. La permeabilidad no es una puerta que la empresa controla unilateralmente; es un conjunto de derechos de conexión y desconexión.

Evidencia de madurez

Las políticas prueban intención. La madurez se observa en eventos. ¿Cuánto tarda una fuente cualificada en ser productiva? ¿Cuántos accesos caducan correctamente? ¿Qué incidentes cambian interfaces? ¿Puede una persona externa disputar una evaluación? ¿Qué porcentaje de salidas deja memoria utilizable? ¿Cuántas capacidades se han cerrado o internalizado por un trigger?

Estas preguntas evitan premiar el discurso. Una empresa puede tener un marketplace interno y seguir dependiendo de favores. Puede documentar cada proceso y bloquear toda iniciativa. Puede usar muchas agencias y carecer de capacidad de acceso porque solo el founder sabe integrarlas.

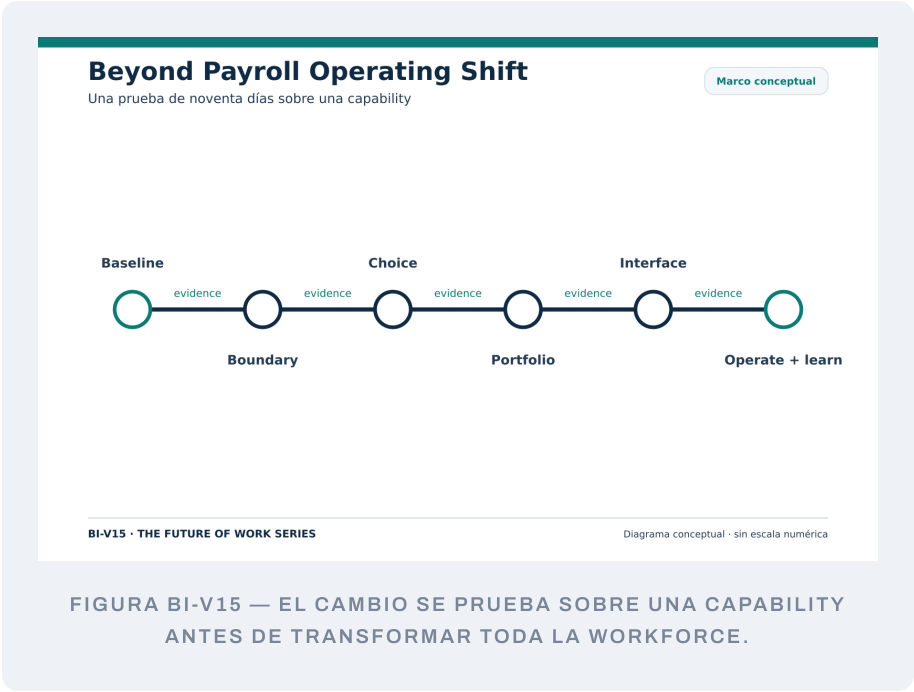
El perfil de permeabilidad debe narrarse con decisiones e incidentes, no con una media. La ausencia de fallos puede significar buen diseño o que nadie los registra.

Madurez no significa máxima apertura. Significa abrir o cerrar la capacidad adecuada con interfaces, evidencia, memoria y responsabilidad suficientes.

1. Open Organization project, **Open Organization Maturity Model**. El framework de este libro utiliza otro nombre, otra unidad de análisis y otra finalidad.

Capítulo 15

BEYOND PAYROLL



De planificar puestos a diseñar capacidades

Vuelva a las dos representaciones del primer capítulo. El organigrama sigue ahí. Continúa mostrando autoridad formal, equipos y relaciones laborales. No lo hemos tachado. A su lado aparece otro mapa: outcomes, capacidades, fronteras, modalidades, interfaces, memoria, riesgo y accountability.

Ahora podemos formular preguntas que el primer mapa no respondía:

- ¿qué trabajo sostiene realmente el outcome?;
- ¿qué fuente es visible y cuál es efectiva?;
- ¿qué capacidad está controlada, comprometida o solo accesible?;
- ¿qué debe sobrevivir a la salida del contributor?;
- ¿quién absorbe variación?;
- ¿dónde queda el conocimiento?;
- ¿qué evento obligará a cambiar la configuración?

Beyond Payroll no significa abandonar el empleo. Significa dejar de usarlo como única unidad intelectual de planificación.

Lo que payroll sigue resolviendo

Una relación laboral puede reunir disponibilidad, aprendizaje, identidad, inversión, autoridad y protección de manera que un conjunto de contratos fragmentados no consigue. En capacidades interdependientes, de conocimiento tácito o alta responsabilidad, puede ser el bundle eficiente.

Nada en este libro demuestra una caída universal del empleo ni una superioridad del trabajo externo. Los datos de BLS sobre *alternative arrangements* muestran una parte relevante y heterogénea del mercado laboral, no un reemplazo total.¹ Las instituciones laborales existen también para distribuir riesgo y derechos. Diseñar “más allá” no autoriza a diseñar alrededor de ellas.

La elección legítima puede ser contratar. La diferencia es que la empresa sabrá qué propiedad de la capacidad lo justifica y qué parte del bundle necesita.

Shift 1: default → choice

El primer cambio ocurre antes de una vacante. Outcome y capability preceden a modalidad. BBBA obliga a comparar build, control estable, borrow y automatización, y a registrar combinación, coste total y trigger.

El éxito no se mide por external share. Se mide por fit:

- demanda y unidad;
- especificidad e interdependencia;
- aprendizaje y apropiación;
- criticidad y recovery;
- disponibilidad de mercado;
- distribución de riesgo.

Una empresa Beyond Payroll no es anti-hiring. Es anti-default.

Shift 2: headcount → portfolio

El segundo cambio amplía el denominador. Horas humanas, gasto y outcomes muestran historias distintas. El Capability Portfolio revela concentración y shocks comunes que una mezcla contractual oculta.

La pregunta deja de ser “¿cuántos externos tenemos?” y pasa a ser:

- ¿qué capacidad depende de una sola fuente?;
- ¿qué fallback es realmente independiente?;
- ¿qué flexibilidad de la firma descansa sobre volatilidad ajena?;
- ¿qué parte de la integración consume al founder?;

Portfolio no convierte personas en activos. Mantiene la fila en la

capability y añade performance bilateral.

Shift 3: organigrama → architecture

El organigrama conserva reporting. La Capability Sourcing Architecture conecta outcome, workflow, decision rights, fuente, interfaz, assurance, memoria y recovery.

Esta segunda vista permite descubrir contradicciones:

- accountability sin acceso;
- automatización sin review;
- proveedor crítico sin transferencia;
- mercado abierto para capacidad urgente;
- empleo estable para una necesidad breve sin aprendizaje;
- permisos permanentes para contribución temporal.

Architecture no sustituye juicio. Hace visible dónde debe ejercerse.

Shift 4: acquisition → orchestration

El match no es el outcome. Sense, Specify, Source, Integrate, Assure, Learn y Reconfigure describen el trabajo que sigue.

La propuesta permanece condicionada. Las rutinas proceden de knowledge integration, resource orchestration y dynamic capabilities; todavía debe probarse que su conjunto aporta explicación incremental. Mientras tanto, su utilidad es práctica: localizar en qué paso se pierde capacidad y contar la carga del núcleo.

Una empresa que amplía fuentes sin reducir Orchestration Load

puede crear un bottleneck humano. El “manager of employees → orchestrator of capabilities” no es promoción retórica. Puede ser una tarea imposible si la organización no invierte en interfaces y memoria.

Shift 5: apertura nominal → permeability

Comprar servicios no vuelve permeable a una empresa. La permeabilidad existe cuando puede abrir y cerrar por capability con gates, permisos, assurance, aprendizaje y salida.

La organización madura no busca Nivel 4 en todo. Elige un optimum contextual. Puede cerrar datos y abrir ideas; controlar decisiones y acceder a ejecución; usar una comunidad y conservar assurance institucional.

El criterio no es “más”. Es “suficiente y gobernable”.

Tres modos de fallo

THE HOLLOW COMPANY

La empresa externaliza ejecución y también pierde capacidad para formular, evaluar, aprender y responder. Mantiene marca y contratos, pero depende de knowledge que no puede juzgar. El Solid Minimum se ha vaciado.

Señales: decisiones sin expertise interno, proveedores autosupervisados, memoria externa, switching imposible.

THE ORCHESTRATION BOTTLENECK

La empresa reúne mucha capacidad y toda pasa por pocos integradores. El founder o un manager se convierte en router de

contexto y aprobación. El leverage aparente se sostiene sobre carga invisible.

Señales: colas de decisión, reuniones para sincronizar especialistas, rebrief constante, horas senior crecientes.

THE REGULATORY FICTION

El contrato llama independiente a una relación de control y dependencia, o la empresa cree que externalizar ejecución externaliza deberes. El sistema optimiza etiquetas y conserva la sustancia.

Señales: prescripción diaria, exclusividad de facto, falta de voz, accountability nominal, datos o permisos sin owner.

Los tres fallos pueden coexistir. Una empresa hueca puede ejercer control excesivo sobre contributors y depender de un founder para integrar.

Una agenda de noventa días

No transforme toda la workforce. Elija una capacidad material, reversible y suficientemente importante para aprender.

DÍAS 1–15: BASELINE Y BOUNDARY

- defina outcome y estándar;
- reconstruya trabajo humano, capital e inputs;
- mapee frontera contractual, accesible y efectiva;
- declare controlled, committed y accessible;
- identifique un denominador engañoso.

DÍAS 16–30: CHOICE

- aplique Separability Test;
- identifique Solid Minimum;
- complete BBBA Decision Record;
- calcule Total Access Cost como rango;
- registre riesgo bilateral y trigger.

DÍAS 31–45: PORTFOLIO

- añada la capacidad a tres vistas: horas, gasto y outcome;
- localice concentración y covarianza;
- pruebe un escenario de shock;
- defina fallback real.

DÍAS 46–60: INTERFACE

- construya Capability Interface Contract;
- separe execution y assurance;
- asigne permisos y caducidad;
- prepare memory y exit.

DÍAS 61–90: OPERATE AND LEARN

- mida access, ramp y capability time;
- registre Orchestration Load;
- observe calidad, errores, learning y bilateral outcomes;
- ejecute una revisión con el trigger;
- decida profundizar, cambiar o cerrar.

El piloto no busca demostrar la tesis. Busca encontrar dónde falla en su empresa.

Una regla para aprobar decisiones

Ninguna decisión material de workforce debería aprobarse sin ocho respuestas:

1. ¿qué outcome?;
2. ¿qué capability?;
3. ¿qué unidad y dependencias?;
4. ¿qué modalidad y alternativa?;
5. ¿qué coste total y riesgo bilateral?;
6. ¿qué Solid Minimum?;
7. ¿quién responde y cómo se recupera?;
8. ¿qué evidencia y fecha harán revisar?

La regla no añade una aprobación central. Puede vivir en un Decision Record de una página. Su valor está en romper el salto automático desde necesidad hasta puesto o herramienta.

¿Solo workforce planning con nombres nuevos?

Strategic workforce planning, total talent management, open innovation, sourcing y organization design ya cubren partes de este terreno. Sería falso presentar la serie como creación ex nihilo.

La contribución conjunta es el cambio de unidad y denominador:

- capability en lugar de categoría contractual;
- tres fronteras en lugar de inside/outside;
- Total Work y Total Access Cost frente a revenue-per-employee;
- Solid Minimum frente a *core* tautológico;
- interface, memory y recovery dentro de sourcing;
- bilateral outcomes dentro de elasticity;
- accountability como centro que no migra con ejecución.

Si esos cambios no mejoran una decisión, el lenguaje no merece sobrevivir. Los frameworks se juzgan por uso, no por novedad nominal.

Red Hat y la organización que no es hueca

La relación entre Red Hat y ecosistemas open source funciona como síntesis precisamente porque no ofrece una empresa sin estructura. Hay empleo, producto, marca, soporte, clientes, governance y comunidades que no pertenecen enteramente a la firma. Valor puede producirse más allá de payroll mientras la institución conserva capacidades para integrar y responder.

El caso tampoco garantiza armonía. Las comunidades tienen intereses, reglas y poder; la empresa captura y redistribuye valor de maneras discutibles. La permeabilidad no elimina política. La hace visible.

La pregunta que queda

Este libro transforma la concepción de workforce. La empresa ya no aparece como una colección de puestos, sino como un sistema de capacidades controladas, comprometidas, accesibles y

automatizadas. Sin embargo, el sistema necesita una institución: propósito, capital, decisiones, memoria, responsabilidad y derecho a comprometerse con otros.

Cuanto más se separan ejecución y contrato, más urgente se vuelve la pregunta por la firma. ¿Por qué existe? ¿Qué frontera queda? ¿Qué significa “líquida” si la responsabilidad no puede licuarse? ¿Qué cambia cuando software e IA no solo ayudan, sino que ejecutan pasos?

La frase que cierra este libro no es una conclusión. Es la puerta del siguiente:

Si cada vez más capacidades pueden producirse fuera de la frontera contractual, ¿qué es exactamente la empresa que las reúne, aprende y responde por ellas?

Un sistema de decisiones, no una política de plantilla

La consecuencia de *Beyond Payroll* no debería ser una meta corporativa sobre porcentaje externo, automatización o headcount. Es un sistema de decisiones que empieza por outcomes y termina en responsabilidad. Las composiciones cambian; el protocolo permanece.

Ese protocolo puede integrarse en cinco ciclos. Estrategia define capacidades que necesitará. Planificación financiera asigna recursos y opciones. People y Procurement diseñan modalidades y relaciones. Tecnología y Riesgo establecen interfaces y assurance. Operaciones observa performance y activa reconfiguration. Ningún ciclo gobierna por separado.

La organización debe resistir la tentación de crear un programa

temporal de “talento flexible”. Si el nuevo lenguaje vive en una oficina y las decisiones reales siguen empezando con puestos, facturas o herramientas, el cambio será cosmético. La capability debe convertirse en objeto compartido de presupuesto, riesgo y aprendizaje.

Qué métricas llegan al comité ejecutivo

Un cuadro reducido puede contener seis señales: *time-to-capability*, Total Access Cost, concentración crítica, Orchestration Load, learning/recovery y Bilateral Performance. No se suman. Cada una responde una pregunta.

Time-to-capability distingue velocidad de match y productividad. Total Access Cost evita comparar salario y tarifa de forma asimétrica. Concentración muestra fragilidad. Orchestration Load revela trabajo invisible. Learning/recovery comprueba si el sistema acumula y sobrevive. Bilateral Performance impide que elasticidad signifique solo traslado de riesgo.

El comité no necesita revisar toda capacidad. Debe observar las materiales, las que cruzan triggers y las que contradicen la estrategia. Una métrica sin decisión asociada se elimina.

La transición organizativa

El piloto de noventa días produce evidencia, no una transformación completa. Después, la empresa puede construir un portfolio inicial de diez a veinte capacidades, elegir tres interfaces recurrentes y definir una revisión semestral. La expansión debe seguir el valor: capacidades críticas, alto gasto externo, automatización material o dependencia de founders.

La secuencia importa. Primero baseline y lenguaje común. Después decisiones y registros. Luego infraestructura de acceso, permisos y memoria. Finalmente governance de portfolio. Intentar desplegar un maturity model completo antes de cambiar una decisión crea burocracia defensiva.

La transición también necesita límites. No reclasificar empleados como contractors para cumplir una meta. No centralizar toda contratación externa. No llamar agente a cada automatización. No exigir a pequeñas tareas el mismo proceso que a capacidades reguladas. La proporcionalidad preserva adopción.

Qué sigue mereciendo empleo

Al finalizar el libro, la contratación deja de ser default y se vuelve una elección positiva. Se emplea cuando continuidad, autoridad, aprendizaje, team production, disponibilidad, protección o identidad institucional forman parte del outcome. Esta formulación es más fuerte que defender empleo por tradición: explica qué valor compra.

También permite diseñarlo mejor. Si la razón es aprendizaje acumulativo, la empresa debe crear experiencias y movilidad. Si es respuesta a excepciones, debe proteger slack y criterio. Si es identidad y cooperación, debe invertir en relaciones. Contratar sin activar esas propiedades compra el bundle y desperdicia su ventaja.

Más allá de payroll no significa menos atención a quienes están en payroll. Una workforce plural aumenta la importancia de una propuesta de empleo clara y de fronteras legítimas.

La tesis frente a sus falsificadores

El argumento perdería valor si las decisiones por capability no mejoraran la comparación de modalidades; si interfaces y orquestación añadieran más coste que el que reducen; si la frontera efectiva no pudiera medirse con suficiente fiabilidad; o si la flexibilidad dependiera sistemáticamente de deteriorar condiciones y transferir riesgo.

También podría ocurrir que las mejores organizaciones ya hagan todo esto bajo otros nombres. En ese caso, el valor del libro sería pedagógico y de integración, no originalidad teórica. Esa posibilidad debe aceptarse. Los conceptos merecen sobrevivir solo si hacen visible una diferencia y cambian una acción.

La serie no necesita que toda empresa se vuelva permeable. Necesita demostrar que, donde las fuentes se multiplican, el diseño consciente supera al default y que la responsabilidad no migra con facilidad.

La pregunta institucional

Después de cambiar la unidad de workforce, aparecen problemas que este libro no puede resolver desde sourcing. ¿Por qué debe existir una entidad estable? ¿Qué derechos, activos y relaciones constituyen su mínimo? ¿Cómo se reconfigura sin perder identidad? ¿Qué ocurre cuando la inteligencia técnica participa en ejecución y decisión? ¿Cuándo una red se convierte en plataforma y quién gobierna sus reglas?

Estas preguntas desplazan el foco desde quién hace el trabajo hacia qué es la empresa que promete el outcome. *The Liquid Company* comienza allí: no con la celebración de una organización fluida, sino con las razones históricas e institucionales por las que la firma fue una solución.

Beyond Payroll no significa menos empleo. Significa elegir la modalidad de cada capacidad con el sistema completo a la vista.

NOTAS DEL CAPÍTULO

1. U.S. Bureau of Labor Statistics, *Contingent and Alternative Employment Arrangements — July 2023*. La publicación mide arreglos en el empleo principal o único y no equivale a todo trabajo externo.

HERRAMIENTA — BEYOND PAYROLL REVIEW

INTERFACE REVIEW

¿Una fuente nueva puede identificar outcome, contexto, inputs, decision rights, permisos, aceptación, escalado, memoria y salida sin depender de una sola persona?

PERMEABILITY REVIEW

Evalúe por capability las once dimensiones. No calcule media total. Declare profile objetivo, evidencia, gap y acción.

90-DAY CAPABILITY SHIFT

Ejecute la agenda del capítulo sobre una sola capacidad. El criterio de éxito no es externalizar ni automatizar: es mejorar la calidad de la decisión y producir evidencia para reconfigurar.

ABIERTA LO SUFICIENTE

Considere cuatro capacidades: estrategia de producto, traducción de catálogo, respuesta a incidentes y una investigación exploratoria. Aplicar el mismo nivel de apertura a las cuatro sería cómodo y absurdo.

Producto puede necesitar aprendizaje cercano y decision rights estables, mientras se abre research. Traducción puede usar mercado amplio con glosario y QA. Incident response necesita reserva, accesos preparados y recovery probado. La investigación puede abrirse a retos o redes sin prometer continuidad operacional.

La organización madura no alcanza un nivel superior común. Construye cuatro perfiles.

Perfil de estrategia de producto

El outcome combina conocimiento de cliente, datos, prioridades y apuestas. Parte del trabajo puede accederse: investigación especializada, diseño, análisis o facilitación. El Solid Minimum probablemente conserva decisión, memoria de trade-offs y relación con clientes.

La interfaz debe permitir challenge, no convertir estrategia en deliverable externo. Assurance no es aprobar una presentación; es comprobar que la decisión incorpora evidencia, restricciones y ownership. El aprendizaje debe quedar en el equipo que decidirá después.

La permeabilidad adecuada abre perspectivas y cierra el derecho final de comprometer producto.

Perfil de traducción

El trabajo puede ser más modular: fuente, idioma, glosario, formato y aceptación. Un market/open set ofrece variedad y escala; una red committed aporta consistencia. Automatización puede producir primer borrador; revisión humana maneja contexto y daño.

La empresa no necesita control estable sobre cada ejecución. Sí necesita estándares, datos protegidos, QA, memoria terminológica y una ruta de escalado. La apertura puede ser alta porque la interfaz absorbe variación.

Si el contenido incluye información regulada o sensible, el perfil cambia. No existe una capacidad llamada “traducción” sin contexto.

Perfil de incident response

La demanda es irregular y el daño temporal alto. El open set puede aportar especialización, pero no garantiza prioridad durante un shock común. La arquitectura necesita core o committed capacity, guardias, permisos preaprobados y una persona interna con autoridad.

Assurance y recovery dominan. La salida de una fuente debe dejar conocimiento del incidente. El perfil puede parecer menos abierto y ser más responsable.

Perfil de investigación exploratoria

Cuando la pregunta está abierta, especificar demasiado destruye valor. Challenges, comunidades o expertos diversos pueden ampliar

inteligencia. La interfaz define pregunta, límites, derechos y forma de evaluación sin imponer método.

El outcome inicial puede ser una solución o un conjunto de hipótesis, no capacidad operacional. Integración ocurre después. Confundir ambas fases conduce a demos que nunca cambian práctica.

La organización no tiene una puntuación

Los cuatro perfiles muestran por qué una media de permeabilidad carece de sentido. Buena apertura en traducción no compensa ausencia de accountability en incidentes. Alto control en producto no convierte a la empresa en cerrada. La unidad de análisis debe permanecer en la capacidad.

El heatmap sirve para localizar gaps. Cada gap recibe evidencia, acción y fecha. Algunas dimensiones quedan deliberadamente bajas. La madurez consiste en poder explicar por qué.

La prueba de cierre

Abra cada capacidad y simule su final. Producto: ¿quién conserva las razones? Traducción: ¿pueden exportarse glosarios y memoria? Incidentes: ¿qué accesos se revocan y qué evidencia permanece? Investigación: ¿qué derechos permiten utilizar la solución?

Una organización que solo diseña entrada acumula deuda. La salida revela si la frontera era gobernable.

La prueba humana

Pregunte también qué vive cada contributor. ¿Comprende el outcome? ¿Puede rechazar o apelar? ¿Recibe contexto y pago

suficientes? ¿La reserva se remunera? ¿La evaluación corresponde a variables controlables? La arquitectura puede ser operacionalmente elegante y depender de relaciones que no se sostienen.

La bilateralidad no prescribe una modalidad. Impide que la empresa declare madurez ignorando el otro balance.

El resultado

“Abierta lo suficiente” significa que cada capacidad accede a variedad sin perder las propiedades que necesita para cumplir, aprender y responder. A veces el diseño abre más; a veces construye compromiso o control. Siempre declara el motivo y el trigger.

Con esta prueba, *Beyond Payroll* deja de ser una invitación a mirar fuera y se convierte en una disciplina para elegir fronteras. La siguiente cuestión ya no pertenece solo a workforce. Pertenece a la institución que mantiene esos perfiles coherentes.

Cinco decisiones que pueden revertirse

La arquitectura propuesta no conduce siempre hacia más apertura. Su madurez se observa en la capacidad de corregir cinco decisiones sin convertir el cambio en derrota.

La primera es **volver a empleo**. Una empresa comienza accediendo a expertise por proyectos. Con el tiempo, el outcome se vuelve recurrente, la persona recibe dirección cotidiana, el conocimiento debe acumularse y la dependencia crece. Mantener la etiqueta externa puede ser operacional y jurídicamente incoherente. Crear un puesto no traiciona *Beyond Payroll*; aplica su principio de correspondencia entre modalidad y trabajo.

La transición exige más que ofrecer un contrato. Debe aclarar rol,

compensación, antigüedad cuando corresponda, propiedad, relaciones existentes y efectos para otras personas. Si la persona no desea o no puede incorporarse, la empresa necesita otra configuración. El objetivo no es capturar al contributor, sino construir la capability responsablemente.

La segunda es **volver a una relación committed**. Una función internalizada puede necesitar variedad o independencia que una plantilla no ofrece. En lugar de externalizar por completo, la empresa crea un partnership estable con capacidad reservada, governance y aprendizaje compartido. Conserva un owner interno y abre parte de la ejecución.

Esta configuración puede aliviar un bottleneck sin perder memoria. También introduce precio por disponibilidad y dependencia relacional. El Decision Record compara ambos lados, no solo el coste salarial anterior.

La tercera es **retirar automatización**. Un workflow puede superar baseline al principio y dejar de hacerlo cuando cambia el dato, la tarea o el comportamiento de usuarios. La inversión hundida no convierte el sistema en capability permanente. Los stop conditions deben permitir bajar a copilot, shadow mode o proceso manual.

Retirar no borra aprendizaje. Los evals, casos de fallo e interfaces pueden mejorar la siguiente versión. La organización comunica el cambio a quienes habían adaptado su trabajo y evita presentar la reversión como culpa de usuarios “resistentes”.

La cuarta es **cerrar una frontera**. Una capability que parecía modular puede revelar alta especificidad, daño o concentración. La empresa reduce providers, restringe permisos o incorpora activos. El cierre debe ser proporcional: controlar datos o decision rights puede bastar sin internalizar toda tarea.

La quinta es **abrir una frontera que la tradición cerró**. Una función histórica puede consumir capital y aprendizaje sin diferenciar. El mercado ha madurado, las interfaces son claras y existen alternativas. Abrir libera atención para el Solid Minimum. El proceso preserva obligaciones laborales, transición y knowledge; no trata el cambio como sustitución instantánea.

Las cinco reversiones comparten una regla: la modalidad no es un juicio sobre el valor de la persona ni una ideología sobre la empresa. Es una hipótesis sobre cómo producir un outcome en un periodo. Al revisar la evidencia, la institución puede cambiar la hipótesis y conservar respeto, memoria y responsabilidad.

La conversación que debe ocurrir antes de aprobar

Imagine una reunión de inversión. El área solicita presupuesto para “contratar un equipo”, “firmar una agencia” o “desplegar agentes”. El lenguaje ya contiene una solución. La primera intervención consiste en volver al outcome: qué debe mejorar, para quién, durante cuánto tiempo y bajo qué estándar.

La segunda dibuja la unidad. ¿Es una tarea, workflow o capability? ¿Qué dependencias y excepciones incluye? ¿Dónde se produce juicio? Una solicitud de cinco roles puede ocultar dos capabilities y un problema de proceso; una herramienta puede resolver una tarea y dejar la capability intacta.

La tercera establece baseline. Volumen, tiempo, calidad, coste total, carga, incidentes y aprendizaje. Si no existe dato, se define un periodo de observación o un rango. La urgencia no autoriza a comparar una propuesta precisa con un presente vacío.

La cuarta genera alternativas reales: Build, Buy, Borrow, Automate

y combinaciones. Cada alternativa incluye Solid Minimum, interfaz, assurance, memoria, recovery y efecto bilateral. Una opción que no puede describirse con estos elementos todavía es una idea, no un plan operativo.

La quinta elige un experimento o compromiso. Para incertidumbre alta y daño bajo, puede ser pequeño y reversible. Para daño alto, la evidencia y control llegan antes. El Decision Record declara la teoría rival y el trigger de revisión.

Esta conversación puede caber en una página si la capability es sencilla. Su propósito no es burocratizar. Impide que una etiqueta de supply capture la decisión antes de comprender el trabajo.

Lo que cambia para People, Procurement y Technology

People deja de recibir solo requisiciones y participa en arquitectura de capabilities. Protege la correspondencia entre relación y realidad, construye pipelines, diseña aprendizaje y observa outcomes humanos en todas las modalidades. No se convierte en owner de cada contractor; ayuda a que la institución no borre personas fuera de payroll.

Procurement deja de optimizar únicamente precio y compliance de alta. Diferencia spot, committed y critical; negocia observabilidad, cambio, memoria y salida; incorpora Total Access Cost y concentración. Su conocimiento de mercados se convierte en capability estratégica.

Technology deja de presentar herramientas como capacidad terminada. Mapea workflows, identity, permisos, datos, evals y fallback. Trabaja con owners de outcome y reconoce que una API

puede ser un proveedor crítico con reglas propias.

Finance crea denominadores comparables. Separa headcount de Total Work, coste nominal de acceso total y margen de dependencia. Puede valorar option, slack y recovery sin fingir una precisión que el dato no permite.

Legal traduce hechos y jurisdicción, no valida etiquetas aisladas. Identifica control, privacidad, propiedad, deberes y recurso. El framework no sustituye su juicio; le ofrece un mapa más fiel de la operación.

Operations integra. Diseña interfaces, cadencias, incidentes y memoria. Su carga debe medirse para que la apertura no dependa de heroísmo. En organizaciones pequeñas, estas funciones pueden residir en pocas personas, pero ninguna desaparece.

Lo que cambia para quien aporta capacidad

Un profesional puede utilizar el Capability Interface Contract para preguntar por outcome, decision rights, disponibilidad, aceptación, feedback, propiedad y salida. La claridad permite valorar si la relación es spot, committed o materialmente otra cosa. También permite cotizar contexto, reserva y rework en lugar de regalar integración.

Una firm especializada puede mostrar cómo conserva continuidad, asegura calidad y transfiere memoria. Puede negociar una relación que recompense inversión en vez de competir siempre por entrega. La independencia no exige anonimato.

Una persona empleada puede ver qué parte de su puesto es capability acumulativa y qué parte puede cambiar. La conversación de carrera se desplaza desde tareas defendidas hacia juicio,

relaciones, integración y aprendizaje. Eso no garantiza seguridad, pero ofrece una base más honesta para diseñarla.

Quienes están afectados por algoritmos o marketplaces necesitan recurso legible. La organización que compra la capability no puede asumir que la plataforma resuelve toda voz. La bilateralidad llega hasta la interfaz donde se asigna, evalúa o termina trabajo.

Una definición de madurez

Una empresa madura no es la que tiene el menor payroll ni el mayor porcentaje externo. Es la que puede seleccionar una capability y mostrar outcome, trabajo total, modalidad, fuentes, owner, derechos, coste, aprendizaje, riesgos, recovery y próxima revisión. Puede explicar por qué esa configuración supera una alternativa y qué evidencia la haría cambiar.

También puede escuchar la experiencia del otro lado y corregir una relación que solo funciona para su balance. Puede internalizar sin vergüenza, externalizar sin abdicar, automatizar sin antropomorfizar y parar sin ocultar el fallo.

La madurez incluye declarar unknowns. Un rango honesto, una muestra limitada o una claim fechada son mejores que una cifra exacta sin denominador. La organización crea capacidad de medición antes de convertir el futuro en objetivo.

Más allá de payroll no hay un destino. Hay un espacio de elección más amplio y una obligación mayor de integrar. El epílogo cierra con una frontera abierta, pero no vacía: la empresa conserva aquello que le permite prometer, aprender y responder, y accede al resto bajo condiciones que puede explicar.

El valor de no decidir todavía

Una arquitectura de opciones también debe reconocer cuándo falta evidencia. La presión por llenar un puesto, firmar un proveedor o adoptar una herramienta puede convertir incertidumbre en compromiso prematuro. “No decidir todavía” es legítimo si tiene propósito, coste, experimento y fecha.

La organización puede separar la parte reversible. Antes de construir una función completa, compra una investigación; antes de externalizar un outcome, prueba una tarea; antes de automatizar, mide baseline; antes de abrir un país, valida demanda y obligación. El experimento reduce una incertidumbre concreta, no simula la operación futura.

Esperar tiene un coste que debe registrarse: oportunidad, carga actual, riesgo y señal al equipo. Si ese coste supera el valor de información, se decide con incertidumbre. La disciplina no promete certeza; hace explícito qué no se sabe.

También existe una no-decisión irresponsable: renovar por defecto, sostener una clasificación dudosa, dejar un permiso abierto o mantener a una persona como bottleneck. La ausencia de acta no significa ausencia de elección. El sistema debe detectar compromisos que se acumulan por inercia.

Un **Option Record** sencillo puede incluir la decisión diferida, el supuesto clave, el experimento, el coste de esperar, la fecha y el owner. Al llegar la fecha, mantener el estado requiere nueva justificación. Así la flexibilidad deja de ser evasión.

Esta práctica completa la lógica *Beyond Payroll*. Elegir modalidad no consiste en responder rápido a un menú. Consiste en preservar opciones hasta que el aprendizaje justifique compromiso y, después,

comprometer con suficiente claridad para que personas y sistema puedan operar. Una frontera gobernada sabe abrir, cerrar y esperar.

Glosario

Accessible capability

Fuente de capacidad visible y potencialmente contratable, sin compromiso suficiente de disponibilidad, contexto o continuidad. Se diferencia de capacidad efectiva: la segunda ya supera gates para un outcome concreto.

Assurance

Conjunto de criterios, pruebas, revisión, observabilidad y escalado que permite aceptar o bloquear un output antes de que cause daño.

Bilateral Performance

Lectura conjunta del outcome para la firma y de la distribución de ingreso, gasto, autonomía, aprendizaje, protección, dependencia y riesgo para quienes contribuyen.

Build / Buy / Borrow / Automate

Sistema de decisión que amplía *Build, Borrow, or Buy* de Capron y Mitchell con automatización, combinaciones, coste total, reversibilidad y outcomes bilaterales.

Capability

Facultad de un sistema para producir un outcome de manera repetible bajo umbrales de calidad, latencia, aprendizaje,

responsabilidad y recuperación. No equivale a skill ni a recurso aislado.

Capability Boundary

Mapa de qué capacidad está bajo control, comprometida, efectiva sin control, solo accesible o fuera de alcance para una capacidad, geografía y fecha.

Capability debt

Obligación futura creada cuando acceso o velocidad se consiguen sin construir memoria, interfaz, assurance, redundancia, accountability u orquestación suficientes.

Capability Interface Contract

Conjunto proporcional de outcome, contexto, inputs, decision rights, permisos, aceptación, feedback, memoria y salida que conecta una fuente con un workflow. No es necesariamente contrato jurídico.

Capability Portfolio

Vista por capacidad de composición, horas, gasto, outcome, compromiso, concentración, aprendizaje, fallback, riesgo y trigger. Las personas no son la unidad del portfolio.

Committed capability

Capacidad externa con reserva, repetición, memoria relacional y expectativas verificables de respuesta, sin asumir que equivalga a empleo.

Controlled capability

Capacidad sobre la que la organización conserva control estable suficiente mediante empleo, propiedad, derechos, contratos, infraestructura o governance.

Effective Capability Boundary

Conjunto de fuentes que pueden producir realmente un outcome específico por encima de umbrales definidos. No es igual al catálogo visible.

Fractional capacity

Unidad parcial y recurrente de expertise, criterio o ejecución. Solo encaja cuando volumen, cadencia, contexto, disponibilidad y accountability admiten fraccionamiento.

Job unbundling / rebundling

Separación de tareas, decisiones o capacidades de un job y diseño posterior de una unidad coherente que recompone excepciones, aprendizaje, derechos y responsabilidad.

Orchestration Loop

Operacionalización propuesta de siete rutinas: Sense, Specify, Source, Integrate, Assure, Learn y Reconfigure. Procede de tradiciones de knowledge integration, resource orchestration y dynamic capabilities.

Outcome

Estado verificable producido para un usuario o sistema, con estándar, plazo, restricciones y externalidades. No equivale a deliverable.

Permeability

Capacidad gobernada de abrir y cerrar flujos de trabajo, conocimiento, datos y decisión a través de fronteras, conservando memory y accountability.

Separability Test

Prueba de modularidad, interdependencia, evaluabilidad, contexto, accountability, aprendizaje y relación/protección antes de separar una unidad de trabajo.

Solid Minimum

Conjunto mínimo de derechos, relaciones, conocimiento, activos, redundancias y rutinas que debe permanecer estable para conservar ventaja, decisión, aprendizaje, recovery y responsabilidad.

Talent Cloud / Access Layers

Talent Cloud es el título narrativo para fuentes accesibles. **Access Layers** es el modelo analítico: Controlled Core, Committed Network y Market/Open Set.

Total Access Cost

Coste de precio/compensación, búsqueda, integración, coordinación, assurance, rework, memoria, riesgo y salida para cualquier modalidad.

Total Work

Trabajo humano atribuible al outcome, incluido empleo, founders, independientes, agencias y proveedores cuando pueda estimarse. Se utiliza con capital e inputs, no como métrica aislada.

Herramientas

1. Capability Baseline

ELEMENTO	REGISTRO
Outcome y estándar	
Empleo directo	
Otro trabajo humano	
Capital/software/datos	
Inputs/proveedores	
Decision owner	
Assurance	
Failure/recovery	

2. Capability Boundary Audit

FUENTE	ESTADO	EVIDENCIA DEL GATE	LATENCIA	MEMORIA	RECOVERY	TRIGGER
	Controlled / Committed / Effective / Accessible / Out					

3. BBBA Decision Record

Outcome · propiedades · separabilidad · Solid Minimum · alternativas · combinación · Total Access Cost · bilateral risk · option preserved · trigger · review date.

4. Fractional Fit Test

DIMENSIÓN	EVIDENCIA	RIESGO / DECISIÓN
Volumen/cadencia		
Modularidad		
Contexto		
Disponibilidad		
Autoridad		
Conflicto		
Regulación		
Aprendizaje		
Owner end-to-end		

5. Separability Test

Mapa de tasks/decisions → dependencias → evaluabilidad → contexto → accountability → aprendizaje → derechos → decisión de separar/mantener/rebundlear.

6. Capability Portfolio

CAPABILITY	OUTCOME	MODALIDAD	HORAS	GASTO	CONCENTRACIÓN	LEARNING	FALLBACK	BILATERAL RISK	TRIGGER

7. Solid Minimum Test

Para Advantage, Residual control, Cumulative learning, System accountability, Bottleneck y Complementarity: objeto mínimo, forma de control, coste de rigidez y trigger.

8. Capability Sourcing Canvas

Outcome → capability → workflow → decision rights → modality/source → interface → assurance → memory/recovery → trigger.

9. Access Layers Map

LAYER	PROFUNDIDAD ELEGIBLE	ACCESS TIME	RAMP TIME	RESERVA	VARIACIÓN	TOTAL COST	CONSTRAINT
Controlled Core							
Committed Network							
Market /Open Set							

10. Orchestration Audit

RUTINA	OUTPUT ESPERADO	EVIDENCIA	FAILURE MODE	ACCIÓN
Sense	Baseline + gap			
Specify	Capability brief			
Source	Decision Record			
Integrate	Interface			
Assure	Acceptance + fallback			
Learn	Cambio de práctica			
Reconfigure	Nueva configuración			

11. Authority–Dependency Matrix

Clasifique cada relación por autoridad y dependencia en ambos sentidos. Allí donde ambas sean altas, revise estatus, derechos, protección, voz, reserva y salida con asesoramiento local.

12. Capability Interface Contract

Outcome · context · inputs/dependencies · decision rights · permissions · acceptance/observability · feedback/escalation · memory/exit.

13. Capability Permeability Review

Heatmap por capability $\times$ demand, modality, access, interface, integration, assurance, knowledge, decision rights, risk, bilateral outcomes y reconfiguration. Sin media total.

Índice de figuras

1. Capability Baseline / Hidden Labor — capítulo 1
2. Capability Usability Gate — capítulo 2
3. Controlled-Committed-Accessible — capítulo 3
4. Capability Sourcing Decision Path — capítulo 4
5. Demand-Unit Fit — capítulo 5
6. Job unbundling and rebundling — capítulo 6
7. Capability Portfolio — capítulo 7
8. Six-Test Solid Minimum — capítulo 8
9. Capability Sourcing Architecture — capítulo 9
10. Four access clocks — capítulo 10
11. Orchestration Loop — capítulo 11
12. Authority-Dependency Matrix — capítulo 12
13. Capability Interface Contract — capítulo 13
14. Capability Permeability Heatmap — capítulo 14
15. Beyond Payroll Operating Shift — capítulo 15

Las especificaciones de producción, fuentes, semantic specs y alternativas accesibles están en la **Visual Bible** de la serie.

Bibliografía

Firma, control y estrategia

- Alchian, Armen A., y Harold Demsetz. “Production, Information Costs, and Economic Organization”. *American Economic Review*, 1972.
- Barney, Jay. “Firm Resources and Sustained Competitive Advantage”. *Journal of Management*, 1991.
- Capron, Laurence, y Will Mitchell. *Build, Borrow, or Buy*. 2012.
- Chandler, Alfred D. Jr. *The Visible Hand*. 1977.
- Coase, Ronald H. “The Nature of the Firm”. 1937.
- Grossman, Sanford J., y Oliver D. Hart. “The Costs and Benefits of Ownership”. 1986.
- Hart, Oliver, y John Moore. “Property Rights and the Nature of the Firm”. 1990.
- Holmström, Bengt, y John Roberts. “The Boundaries of the Firm Revisited”. 1998.
- Kogut, Bruce, y Udo Zander. “Knowledge of the Firm...”. 1992.
- Lepak, David, y Scott Snell. “The Human Resource Architecture”. 1999.
- Williamson, Oliver E. “Transaction Cost Economics: The Natural Progression”. 2009.

Redes, aprendizaje e interfaces

- Argote, Linda. *Organizational Learning*.
- Baldwin, Carliss, y Kim Clark. *Design Rules*.
- Bechky, Beth. “Gaffers, Gofers, and Grips”.
- Benkler, Yochai. “Coase’s Penguin, or, Linux and the Nature of the Firm”. 2002.
- Chesbrough, Henry. *Open Innovation*.
- Goldfarb, Avi, y Catherine Tucker. “Digital Economics”. 2019.
- Powell, Walter W. “Neither Market nor Hierarchy”. 1990.
- Teece, David, Gary Pisano y Amy Shuen. “Dynamic Capabilities and Strategic Management”. 1997.

Trabajo, datos e instituciones

- Autor, David. “Why Are There Still So Many Jobs?”. 2015.
- BLS. *Contingent and Alternative Employment Arrangements — July 2023*.
- Census Bureau. *Nonemployer Statistics*.
- Freeland, Edward, Andrew Garin y Dmitri Koustas. “Costs and Benefits of Freelance Work”. 2026.
- OIT. *Digital labour platforms*.
- Unión Europea. *Directiva (UE) 2024/2831*.

IA relevante para Book I

- Brynjolfsson, Erik, Danielle Li y Lindsey Raymond. “Generative AI at Work”.
- Dell’Acqua, Fabrizio et al. “Navigating the Jagged Technological Frontier”.
- OIT. *Generative AI and Jobs: A Refined Global Index*. 2025.

La **Source Library** del paquete contiene la bibliografía completa, función probatoria y asignación por capítulo.

SOBRE EL LIBRO

Las empresas siguen planificando por puestos, aunque sus resultados dependan de una capacidad repartida entre empleados, especialistas independientes, firmas asociadas, proveedores, software e inteligencia artificial. *Más allá de la nómina* ofrece un método para acceder a esa capacidad —controlada, comprometida o accesible— sin confundir acceso con disponibilidad ni flexibilidad con transferencia de riesgo.

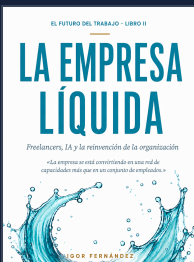


SOBRE EL AUTOR

Igor Fernández

Trabaja con fundadores y equipos directivos para convertir organizaciones dependientes de una persona en empresas que operan, deciden y se adaptan por sistema.

OTROS TÍTULOS DEL AUTOR



La Empresa Líquida

Freelancers, IA y la reinención de la organización.



Empresas que funcionan sin el fundador

Construir, validar y escalar mediante hitos de madurez.