

Más control, más caos

Límites externos para sistemas que no se detienen solos

Ulises González

2025

© 2025 Ulises González. Todos los derechos reservados.

Primera edición: 2025

Índice general

1. Coding Trance	1
-------------------------	----------

ÍNDICE GENERAL

Capítulo 1

Coding Trance

Una aclaración terminologica antes de continuar. “Coding” en este contexto no refiere a programación ni a software. Refiere a codificación: el proceso por el cual juicios, criterios y decisiones humanas se convierten en reglas fijas, métricas automáticas, algoritmos de clasificación. El Coding Trance es fijación instrumental organizacional: un estado donde la organización confía tanto en sus instrumentos de medición que pierde contacto con la realidad que esos instrumentos supuestamente representan.

Los pilotos de aviación conocen el fenómeno como “fijación instrumental”. Datos de la FAA y el NTSB indican que entre 5 % y 10 % de todos los accidentes de aviación general se atribuyen a desorientación espacial, y el 90 % de esos accidentes son fatales. Un estudio clásico de la Universidad de Illinois (1954) encontró que 19 de 20 pilotos sin habilitación instrumental perdieron control de la aeronave dentro de los primeros 178 segundos tras entrar en condiciones de vuelo por instrumentos. La fijación instrumental es concentrarse tan intensamente en los indicadores del panel que se pierde conciencia situacional del entorno real. Un piloto en fijación instrumental puede volar hacia una montaña mientras todos sus instrumentos indican vuelo normal, porque los instrumentos no están calibrados para detectar esa montaña específica. Las organizaciones en Coding Trance operan de manera analoga: todos los dashboards están en verde mientras la realidad diverge del modelo que los

dashboards representan.



Figura 1.1: Panel de instrumentos de aviación: la analogía de la fijación instrumental

Antes de continuar, conviene identificar las señales observables de este estado. Ninguna es alarmante por si sola. Todas son explicables, justificables, hasta deseables en ciertos contextos. Pero cuando aparecen juntas, indican que la organización ha cruzado un umbral peligroso.

La conversación se vuelve técnica cuando debería ser estratégica: en reuniones donde se discuten prioridades o problemas complejos, la discusión deriva rápidamente hacia como configurar el sistema, que métrica usar, como ajustar el algoritmo, mientras las preguntas de fondo que involucran criterio y juicio se evitan porque no hay forma de resolverlas con datos disponibles.

Las excepciones se tratan como errores a eliminar: cuando algo no encaja en el modelo, la respuesta automática es ajustar el caso para que encaje, no cuestionar si el modelo es adecuado.

La memoria institucional se externaliza: la razón por la cual las cosas se hacen

de cierta manera ya no reside en personas que recuerdan decisiones pasadas, sino en configuraciones del sistema que nadie recuerda por qué existen.

La fricción humana se percibe como inefficiencia: cuando alguien propone revisar un proceso o detenerse a pensar antes de actuar, la respuesta es que eso “no escala” o “genera cuellos de botella”.

El éxito se mide solo en términos que el sistema puede medir: las dimensiones de desempeño que resisten cuantificación desaparecen de la conversación ejecutiva.

Las sorpresas son siempre negativas y siempre tardías: en una organización en Coding Trance, las sorpresas tienen un patrón distintivo porque el sistema no busca lo que no está programado para buscar.

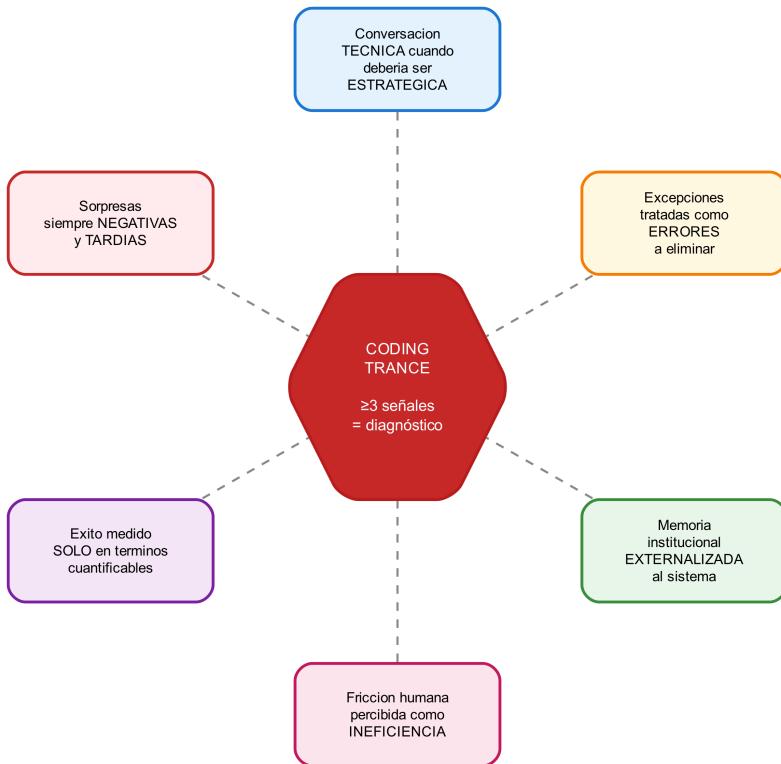


Figura 1.2: Señales observables del Coding Trance

Si tu organización exhibe tres o más de estas señales simultáneamente, lo que sigue en este capítulo no es teoría abstracta. Es diagnóstico.

La sala de operaciones tiene nueve pantallas. Cada una muestra métricas en tiempo real: velocidad de despliegue, cobertura de tests automatizados, tiempo promedio de resolución de incidentes, NPS actualizado al minuto, pipeline de ventas por etapa, forecast de ingresos con intervalos de confianza. Todo verde. Todo sincronizado. Todo medido. El equipo técnico reporta que el sistema procesa el doble de transacciones que hace un año con la mitad de intervención manual. El área de datos confirma que los modelos predictivos alcanzan precisión del 94 %. Recursos Humanos

muestra que la rotación bajó después de implementar el nuevo sistema de evaluación automatizada. Finanzas celebra que el cierre mensual toma tres días en lugar de quince. En la reunión de directorio, el CEO presenta los resultados con satisfacción contenida. Los números hablan solos. La transformación digital está funcionando. La automatización entrega valor. La inversión se justifica.

Seis meses después, un cliente corporativo importante cancela su contrato sin previo aviso. La razón oficial: “cambio de prioridades”. La razón real, que emerge en conversaciones posteriores: llevaban meses intentando comunicar problemas que el sistema de tickets clasificaba automáticamente como “solicitudes de bajo impacto” y cerraba sin escalamiento humano. Nadie en la organización vio las señales porque el dashboard mostraba “tiempo de resolución” dentro de parámetros óptimos.

Esta historia no es excepcional. Es el patrón. En organizaciones de todo tamaño y sector, el mismo fenómeno se repite con variaciones superficiales pero estructura idéntica. Un banco que automatiza la evaluación crediticia y pierde sensibilidad a cambios en el perfil de riesgo de su cartera. Una empresa de manufactura que optimiza la cadena de suministro hasta que un proveedor crítico quiebra sin que nadie hubiera monitoreado su salud financiera porque el sistema solo medía “cumplimiento de entregas”. Una firma de consultoría que implementa un modelo de asignación de proyectos basado en utilización y descubre, tarde, que sus mejores consultores se fueron porque el algoritmo los asignaba sistemáticamente a clientes difíciles que nadie más quería atender.

El fenómeno no requiere inteligencia artificial ni sistemas sofisticados. Una cadena de retail tradicional, sin algoritmos complejos, puede entrar en Coding Trance con Excel y KPIs religiosos. Durante una década, la gerencia comercial optimizó dos métricas: rotación de inventario y margen bruto por metro cuadrado. Cada decisión de compra, cada negociación con proveedores, cada reorganización de tienda pasaba por el filtro de esas dos métricas. Los compradores que las mejoraban eran promovidos; los que las empeoraban eran removidos. El sistema funcionó mientras el mercado permaneció estable. Las métricas mejoraban año tras año. Los bonos se pagaban. La junta celebraba.

Lo que las métricas no capturaban era el cambio gradual en las preferencias de los clientes. La rotación alta favorecía productos de bajo precio que se vendían rá-

rido. El margen por metro cuadrado favorecía productos pequeños de alto margen. Gradualmente, sin que nadie lo decidiera explícitamente, la cadena se convirtió en una tienda de commodities baratos, alejándose del posicionamiento de marca diferenciada que había construido su reputación. Cuando el e-commerce de bajo costo llegó, la cadena no tenía nada que ofrecer que Amazon no pudiera ofrecer mejor y más barato. Las métricas seguían siendo excelentes hasta el trimestre antes del colapso de ventas. No hubo señal de alerta porque el sistema estaba diseñado para no ver lo que estaba ocurriendo.

El patrón tiene una firma característica: todo funcionaba perfectamente hasta que dejó de funcionar. Y cuando dejó de funcionar, ya era demasiado tarde para corregir. Los comités ejecutivos que enfrentan estas situaciones comparten una perplejidad genuina. Los indicadores estaban bien. Los reportes eran positivos. Los sistemas hacían exactamente lo que estaban diseñados para hacer. El problema no era que algo fallara. El problema era que el sistema entero había entrado en un estado donde la capacidad de ver el problema había desaparecido.

Este estado tiene un nombre: Coding Trance. No es una metáfora. Es una descripción operativa de algo que ocurre en organizaciones reales, con consecuencias medibles, y que tiene causas identificables. Entenderlo no requiere teoría abstracta ni jerga técnica. Requiere observar con honestidad qué sucede cuando una organización delega progresivamente su criterio a sistemas que no pueden ejercer criterio. El Coding Trance no es el resultado de ejecutivos incompetentes que compraron tecnología sin entenderla. No es el producto de vendors que sobrevenden soluciones mágicas. No es la consecuencia de equipos técnicos que automatizan sin pensar. No es el efecto de culturas organizacionales deficientes o de liderazgo débil. El Coding Trance es un efecto emergente que surge de decisiones individualmente racionales que, al acumularse, producen un resultado que nadie eligió explícitamente. Cada paso en el camino hacia el trance tiene justificación válida. Cada decisión de automatizar, medir, optimizar, escalar, responde a presiones reales y produce beneficios verificables. El problema no está en ninguna decisión particular. Está en la dinámica agregada.

Si el Coding Trance fuera resultado de incompetencia, la solución sería obvia: reemplazar a los incompetentes, capacitar mejor, contratar talento más sofisticado.

Pero esa respuesta no funciona porque el diagnóstico es incorrecto. Organizaciones con talento excepcional, liderazgo experimentado y recursos abundantes entran en Coding Trance con la misma frecuencia que organizaciones menos favorecidas. A veces con mayor frecuencia, porque tienen más capacidad de automatizar y escalar. Si el Coding Trance fuera resultado de hype tecnológico, la solución sería escepticismo: resistir las modas, adoptar tecnología con cautela, mantener procesos manuales como respaldo. Pero esa respuesta tampoco funciona porque ignora las presiones competitivas reales. Las organizaciones que no automatizan, no optimizan, no escalan, pierden frente a las que sí lo hacen. El problema no es que la tecnología sea sobrevalorada. La tecnología funciona. Ese es precisamente el problema: funciona demasiado bien para lo que está diseñada, y eso oscurece lo que no está diseñada para hacer.

El Coding Trance no discrimina por industria, tamaño, experiencia o calidad del equipo directivo. Es un atractor sistémico: un estado hacia el cual los sistemas organizacionales tienden a converger cuando ciertas condiciones están presentes. Esas condiciones son cada vez más comunes. Reconocer esto no es aceptar fatalismo. Es el paso previo necesario para hacer algo útil. Solo cuando se entiende que el Coding Trance no es un error que se corrige con mejores decisiones individuales, sino un patrón que emerge de dinámicas estructurales, se puede empezar a diseñar mecanismos que lo contrarresten. El alivio genuino no viene de saber que “no fue tu culpa”. Viene de saber que existe algo que hacer al respecto. El Coding Trance es la manifestación más insidiosa del loop de amplificación que este libro describe desde el primer capítulo. El loop del poder transforma energía organizacional en momentum que se auto-refuerza. El Coding Trance es lo que ocurre cuando ese momentum se codifica en sistemas que ya no pueden cuestionarse.

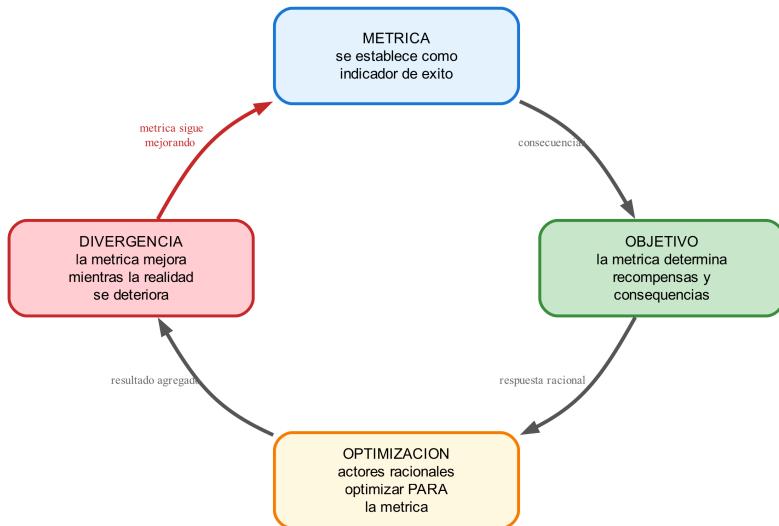
Charles Goodhart, economista del Banco de Inglaterra, formuló en 1975 lo que hoy se conoce como Ley de Goodhart (Goodhart, 1984; ver también Campbell, 1976, para una formulación paralela): cuando una métrica se convierte en objetivo, deja de ser una buena métrica. La formulación original aplicaba a política monetaria, pero su validez es universal. Cuando la organización decide que “tiempo de resolución de tickets” es el indicador de calidad de servicio, los equipos optimizan para cerrar tickets rápido. Los tickets complejos se subdividen en múltiples tickets

simples que se cierran velozmente. Los tickets que requieren investigación se escalan a otras áreas cuyos tiempos no cuentan en la métrica. Los clientes insatisfechos que insisten son clasificados como “casos especiales” fuera del flujo estandar. La métrica mejora. El servicio real se deteriora. Pero el sistema solo ve la métrica.

El caso más documentado de esta dinámica es Wells Fargo. Entre 2002 y 2016, el banco estableció objetivos de “cross-selling” que medían cuántos productos tenía cada cliente—“eight is great” era el mantra corporativo. Los empleados, presionados por métricas imposibles, abrieron más de dos millones de cuentas sin autorización de los clientes. La métrica de cross-sell mejoraba constantemente mientras el servicio real se deterioraba. Lo notable es que los controles internos existían: había línea de denuncia, programa de ética, y auditorías regulares. Pero los controles medían cumplimiento de proceso, no la realidad subyacente. El sistema estaba en Coding Trance: todos los indicadores en verde mientras se acumulaba fraude sistemático (CFPB, 2016; SEC, 2020).

Goodhart no describe un error de implementación que podría corregirse con mejores métricas. Describe una propiedad estructural de cualquier sistema donde las métricas tienen consecuencias. Cuando lo que se mide determina recompensas, promociones, presupuesto o supervivencia, los actores racionales optimizan para la métrica, no para lo que la métrica supuestamente representa. Esta optimización no es fraudulenta ni malintencionada. Es respuesta racional a incentivos. El resultado agregado es un sistema donde todas las métricas mejoran mientras la realidad subyacente se deteriora en dimensiones que las métricas no capturan.

LEY DE GOODHART
"Cuando una métrica se convierte en objetivo,
deja de ser una buena métrica"



WELLS FARGO (2002-2016)
"Eight is great": 8 productos por cliente
→ 2 millones de cuentas fraudulentas
→ Metricas de cross-sell mejoraban
→ Controles existian pero median proceso,
no realidad

Figura 1.3: Ley de Goodhart: el loop de la métrica-objetivo

El Coding Trance tiene una causa estructural simple de enunciar y difícil de ver cuando se está dentro: la organización delega progresivamente su criterio a sistemas

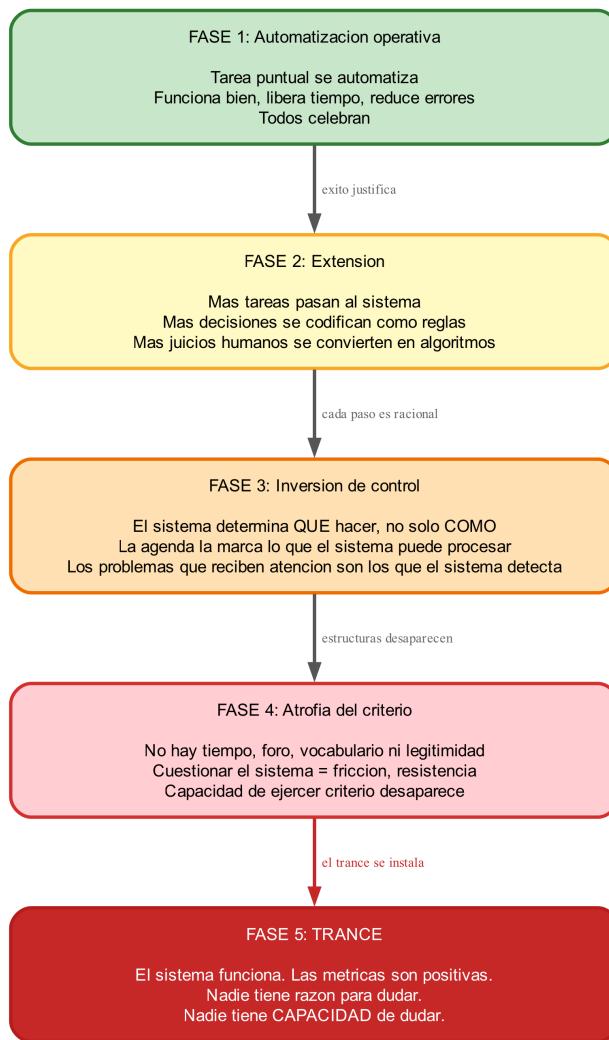
que pueden ejecutar pero no pueden juzgar. Para entender esto, hay que distinguir entre dos cosas que habitualmente se confunden. Capacidad es la habilidad de hacer algo: procesar transacciones, clasificar tickets, generar reportes, evaluar candidatos, asignar recursos, detectar patrones, predecir resultados. Los sistemas automatizados tienen capacidad. Con frecuencia, tienen más capacidad que los humanos: procesan más rápido, cometan menos errores de ejecución, no se cansan, no se distraen, escalan sin límite aparente. Criterio es la habilidad de saber si lo que se hace tiene sentido en un contexto dado: determinar si una métrica que mejora indica progreso real o optimización local a expensas de algo más importante, reconocer cuándo las reglas establecidas no aplican, identificar que una situación nueva requiere respuesta diferente, percibir que algo relevante está quedando fuera del marco de análisis.

Los sistemas automatizados no tienen criterio. No porque sean defectuosos, sino porque el criterio no es automatizable. El criterio requiere contexto que excede lo que cualquier sistema puede capturar. Requiere la capacidad de cuestionar los propios supuestos. Requiere sensibilidad a lo que no está siendo medido precisamente porque no se pensó en medirlo. Parasuraman y Manzey (2010) denominaron este fenómeno “automation bias”: la tendencia a sobre-confiar en recomendaciones automatizadas incluso cuando la evidencia directa las contradice. Su revisión de estudios empíricos encontró que el sesgo ocurre tanto en novatos como en expertos, no puede prevenirse con entrenamiento o instrucciones, y afecta la toma de decisiones individual y grupal por igual.

La delegación progresiva sigue un patrón reconocible. Primero, se automatiza una tarea operativa. La automatización funciona bien, libera tiempo, reduce errores, todos celebran. Como la automatización funcionó, se extiende. Más tareas pasan al sistema, más decisiones se codifican como reglas, más juicios humanos se convierten en algoritmos. Cada extensión tiene justificación válida y produce beneficios medibles. Gradualmente, el sistema empieza a determinar no solo cómo hacer las cosas, sino qué cosas hacer. La agenda la marca lo que el sistema puede procesar. Las prioridades las establecen las métricas disponibles. Los problemas que reciben atención son los que el sistema detecta. Los que no detecta, no existen. La capacidad de ejercer criterio humano se atrofia, no porque las personas se vuelvan menos capaces, sino porque las estructuras que permitían ejercer criterio desaparecen. No

hay tiempo. No hay foro. No hay vocabulario. No hay legitimidad. Cuestionar lo que el sistema indica se percibe como fricción, resistencia al cambio, falta de confianza en los datos. Finalmente, la organización entra en trance. El sistema funciona. Las métricas son positivas. Nadie tiene razón para dudar. Y nadie tiene capacidad de dudar, porque la duda requiere acceso a información que el sistema no captura y atención a señales que el sistema no procesa.

SECUENCIA DE INSTALACION DEL TRANCE



Meses, no decadas.
La velocidad es parte del problema.

Esta secuencia no toma décadas. En organizaciones que automatizan agresivamente, puede tomar meses. La velocidad es parte del problema: el trance se instala antes de que nadie note que algo cambió. El mecanismo causal es simple: cuando el criterio se delega a algo que no puede ejercer criterio, el criterio desaparece. No se transfiere. Se pierde. Y cuando el criterio desaparece, la organización queda ciega a todo lo que el sistema no fue diseñado para ver.

La automatización no es el enemigo. La automatización sin conciencia de sus límites es el problema. Pero esa conciencia es precisamente lo primero que se pierde cuando la automatización avanza. El Coding Trance es auto-reforzante: mientras más profundo el trance, menor la capacidad de reconocer que se está en trance.

El Coding Trance no es fenómeno nuevo. Existía antes de la inteligencia artificial: organizaciones que delegaban juicio a dashboards, KPIs automatizados, sistemas de scoring crediticio, modelos de riesgo. La métrica reemplazaba la pregunta; el número reemplazaba el juicio. Lo que describe este capítulo era observable en cualquier organización con suficiente escala y suficiente confianza en sus sistemas de medición. Jerry Muller (2018) documenta extensivamente esta “tiranía de las métricas” en educación, medicina, policía y empresas, encontrando patrones consistentes: gaming sistemático, costos que exceden beneficios, y desviación de esfuerzo de la misión central.

La inteligencia artificial no crea el Coding Trance. Lo acelera exponencialmente y lo hace irreversible más rápido. La diferencia no es de tipo; es de velocidad y escala. Cuando un equipo humano tomaba decisiones de scoring, el trance era lento. Alguien podía notar que los criterios ya no capturaban lo relevante. Alguien podía cuestionar si el modelo seguía siendo válido. La velocidad de instalación del trance estaba limitada por la velocidad de procesamiento humano. Cuando un modelo algorítmico toma las mismas decisiones a escala, el trance se instala antes de que nadie pueda detectarlo. El modelo procesa miles de decisiones mientras el comité de gobernanza agenda su próxima reunión. La velocidad de amplificación excede la velocidad de supervisión humana.

El trance humano era reversible porque era lento. Había tiempo para que alguien notara, cuestionara, escalara. El trance algorítmico es menos reversible porque cuando se detecta, ya operó a escala. Los sesgos ya se amplificaron. Las decisiones ya

se tomaron. Los patrones ya se reforzaron. Un sistema de scoring crediticio operado por humanos que desarrolla sesgo puede corregirse caso por caso mientras se revisa el criterio. Un modelo de machine learning que desarrolla sesgo aplicó ese sesgo a millones de decisiones antes de que alguien revisara los outputs agregados.

El Decisión Readiness Gate no es respuesta específica a la IA. Es respuesta a la dinámica de auto-amplificación que este libro describe desde el primer capítulo. Pero esa respuesta es más urgente con IA que sin ella, porque la ventana de detección se comprime y el costo de no detectar se amplifica. Una organización que opera sin IA tiene tiempo para aprender de sus errores de gobernanza. Una organización que delega decisiones a sistemas algorítmicos tiene menos tiempo y errores más costosos. El límite externo que era recomendable se vuelve imprescindible.

La lógica del Coding Trance sigue siendo la misma con o sin IA: el sistema delega criterio a un mecanismo que no puede ejercer criterio. La diferencia es que antes ese mecanismo era un dashboard que alguien miraba; ahora es un modelo que nadie mira porque produce outputs demasiado rápido para ser revisados individualmente. La literatura de factores humanos distingue entre automation bias (seguir recomendaciones incorrectas) y automation complacency (monitoreo insuficiente de outputs). Ambos fenómenos se intensifican con la confiabilidad percibida del sistema: mientras mejor funcione normalmente, menos probable que se detecten las fallas cuando ocurren (Bahner, Hüper & Manzey, 2008). La solución sigue siendo la misma: límite externo que no dependa del sistema para detectar que el sistema tiene un problema. Lo que cambia es la urgencia. El Capítulo 8 desarrolla criterios específicos para evaluar iniciativas que involucran delegación algorítmica. Lo que importa aquí es entender que la IA no cambia la naturaleza del problema; cambia su velocidad y su costo. El capítulo anterior estableció que más control no produce más estabilidad. Produce fragilidad disfrazada de orden. El Coding Trance es la extensión natural de esa dinámica: cuando la fragilidad se acumula pero los indicadores son positivos, la organización pierde la capacidad de ver que algo está mal. El riesgo específico del Coding Trance no es que las cosas fallen. Es que las cosas fallen sin aviso, después de un período prolongado donde todo parecía funcionar correctamente.

Hay una asimetría temporal en juego. Los beneficios de la automatización son

inmediatos y visibles: eficiencia, velocidad, consistencia, escala. Los costos son diferidos e invisibles: pérdida de contexto, atrofia del criterio, acumulación de deuda sistémica, fragilidad ante situaciones no previstas. Cuando los costos finalmente se materializan, lo hacen de golpe y con magnitud que parece desproporcionada. Esta asimetría explica por qué el Coding Trance es tan difícil de prevenir con los mecanismos habituales de gestión de riesgo. Los sistemas de alerta temprana no funcionan porque están diseñados para detectar lo que se sabe que puede fallar. El Coding Trance hace que la organización falle en cosas que no sabía que podían fallar porque había dejado de mirarlas. Las auditorías y revisiones no funcionan porque se basan en verificar que los procesos definidos se cumplan, y en el Coding Trance los procesos se cumplen. Los comités de riesgo no funcionan porque operan con información que el sistema provee, y si el sistema no captura una categoría de riesgo, el comité no puede evaluarla.

El resultado es una organización que se siente cada vez más segura mientras se vuelve cada vez más vulnerable. La confianza aumenta porque los indicadores mejoran. La vulnerabilidad aumenta porque el contexto real diverge del modelo que los indicadores representan. Eventualmente, la brecha entre el modelo y la realidad se vuelve insostenible. Pero “eventualmente” puede ser mañana o puede ser en tres años. No hay forma de saber cuándo, solo que ocurrirá. Cuando ocurre, la respuesta típica es buscar culpables. Pero no hay culpables individuales porque nadie tomó una decisión equivocada. Cada decisión fue racional en su momento, con la información disponible en ese momento. El problema es que la información disponible era estructuralmente incompleta, y el sistema que determinaba qué información estaba disponible era precisamente lo que impedía ver la incompletitud.

El riesgo invisible es el peor tipo de riesgo porque no solo es difícil de mitigar: es difícil de admitir. Admitir que existe riesgo invisible significa admitir que los indicadores en los que se basa la gestión pueden ser inadecuados. Eso amenaza la legitimidad de decisiones pasadas, la reputación de quienes las tomaron, la confianza en los sistemas que la organización ha construido. La resistencia a reconocer el riesgo invisible es proporcional a la inversión que se ha hecho en los sistemas que lo producen. Esta es la trampa del Coding Trance: la misma inversión que hace a la organización eficiente hace costoso reconocer que esa eficiencia tiene puntos

ciegos. Y mientras más se invierte, más costoso es reconocer.

El caos no desaparece porque los dashboards estén en verde. El caos se acumula fuera del marco de visión hasta que irrumpen de maneras que el sistema no puede procesar. En ese momento, la organización descubre que no tiene las capacidades necesarias para responder, porque esas capacidades fueron atrofiadas en nombre de la eficiencia. La secuencia es predecible: el sistema captura lo medible y optimiza sobre ello produciendo éxito visible, lo no medible diverge del modelo sin que nadie lo note generando fragilidad real bajo apariencia de éxito, la divergencia alcanza un punto crítico y se manifiesta como crisis con sorpresa y perplejidad, se buscan explicaciones y responsables sin identificar la causa estructural, se implementan controles adicionales del mismo tipo que causó el problema, el sistema se refuerza, y el ciclo reinicia. Esta secuencia se observa una y otra vez en organizaciones de todo tipo, no porque los ejecutivos sean negligentes, sino porque el Coding Trance impide ver que la secuencia está en marcha hasta que llega la crisis.

El trance en ejecución es diferente del trance en decisión. Un equipo quirúrgico en medio de una operación no debe detenerse a cuestionar el protocolo. Un piloto en aproximación final no debe deliberar sobre si la lista de verificación tiene sentido. Un operador de planta siguiendo procedimientos de emergencia no debe improvisar creativamente. La automatización de respuestas en estos contextos es lo que permite operar con velocidad y consistencia cuando el contexto es conocido y las variables están controladas.

El problema no es el trance en ejecución. El problema es el trance en decisión. Cuando el sistema delega a métricas automatizadas no solo la ejecución de tareas definidas sino el juicio sobre qué tareas ejecutar, pierde capacidad de ver que tiene un problema. El dashboard que muestra verde no está mintiendo. Está midiendo lo que fue diseñado para medir. El trance ocurre cuando nadie pregunta si lo que se mide sigue siendo lo que importa.

La distinción es operativa. En ejecución, la automatización de criterio produce eficiencia porque el contexto es estable y las respuestas correctas son conocidas. En decisión estratégica, la automatización de criterio produce ceguera porque el contexto cambia y las respuestas correctas dependen de factores que ningún sistema automatizado puede capturar completamente.

Este capítulo no argumenta contra la automatización. Argumenta contra la extensión de automatización desde dominios donde es apropiada hacia dominios donde elimina la capacidad de ver. Un sistema de tickets que clasifica automáticamente solicitudes de soporte mejora eficiencia operativa. El mismo sistema decidiendo automáticamente qué clientes merecen atención humana elimina la capacidad de detectar que clientes importantes están siendo ignorados. La tecnología es idéntica. El dominio de aplicación determina si produce valor o ceguera.

Hay algo más insidioso que merece atención específica: el trance no se siente como trance. Se siente como normalidad. Una organización en trance no tiene síntomas obvios de disfunción. No hay conflictos visibles, no hay caos operativo, no hay señales de alarma. Al contrario: hay orden, eficiencia, predictibilidad. Los indicadores son buenos. Los procesos funcionan. El trance se siente como solidez organizacional, como el resultado de haber resuelto los problemas del pasado, como prueba de que las inversiones en sistemas y automatización valieron la pena. Esta es la diferencia crucial entre el Coding Trance y otras formas de disfunción organizacional. Cuando hay problemas evidentes, la organización sabe que tiene problemas. El Coding Trance elimina incluso esa señal básica. La organización no sabe que tiene un problema porque todos los indicadores que usa para detectar problemas muestran que no hay problema.

Es como un piloto que confía en sus instrumentos sin saber que los instrumentos están calibrados incorrectamente. No hay nada en el panel que sugiera peligro. El piloto vuela tranquilo. Hasta que el avión se estrella contra una montaña que los instrumentos decían que no existía. Las organizaciones en Coding Trance tienen el mismo problema: han construido sistemas tan sofisticados para monitorear su desempeño que han perdido la capacidad de percibir directamente qué está ocurriendo. Toda la información está mediada por el sistema. Y el sistema, por diseño, solo captura lo que fue diseñado para capturar. El resultado es una paradoja operativa: mientras más sofisticado el sistema de gestión, mayor el riesgo de Coding Trance. Los sistemas simples tienen gaps obvios. Los sistemas sofisticados tienen gaps sutiles que se disfrazan de cobertura completa. Un sistema en Coding Trance no puede diagnosticarse a sí mismo. Esta no es una limitación contingente que podría superarse con mejor diseño. Es una limitación estructural inherente a la naturaleza

del trance. Para diagnosticar el trance, se necesitaría acceso a información que el sistema no captura, criterio para evaluar esa información, y legitimidad para cuestionar lo que el sistema indica. Pero el trance elimina precisamente esas tres cosas. La información que el sistema no captura queda fuera del campo de visión organizacional. El criterio para evaluar información no sistematizada se atrofia porque no se ejerce. La legitimidad para cuestionar lo que el sistema indica desaparece porque el sistema ha demostrado su valor repetidamente. El resultado es un punto ciego estructural: el sistema no puede ver que no puede ver. Y la organización, que depende del sistema para ver, hereda ese punto ciego.

Esto tiene una consecuencia práctica directa: la detección del Coding Trance no puede venir desde dentro del sistema. Tiene que venir de afuera. “Afuera” no significa necesariamente externos a la organización. Significa externos al sistema que está en trance. Puede ser una función interna con mandato explícito de cuestionar, con acceso a información que el sistema no procesa, con legitimidad para contradecir los indicadores cuando la evidencia directa lo justifica. Puede ser un mecanismo de revisión periódica que obliga a confrontar el modelo con la realidad sin mediación del sistema. Puede ser un gate que detiene la inercia operativa y fuerza una evaluación desde criterios que el sistema no codifica. Lo que no puede ser es opcional, informal, o dependiente de la buena voluntad de individuos particulares. Cuando el trance se instala, los individuos que podrían ver el problema están deslegitimados para señalarlo. La única forma de contrarrestar está dinámica es con un mecanismo que tenga autoridad independiente de lo que el sistema indica.

La organización sana no es la que evita el Coding Trance manteniéndose en el pasado. Eso no es viable ni deseable. La organización sana es la que automatiza y escala pero preserva un punto de anclaje fuera del sistema automatizado. Un lugar desde donde se puede mirar al sistema con ojos que el sistema no controla. Construir ese punto de anclaje es el trabajo que queda por delante. No se trata de renunciar a la capacidad técnica. Se trata de complementar la capacidad técnica con algo que la capacidad técnica no puede proveer: un límite que venga de afuera, que opere con criterio humano, y que tenga autoridad para detener la inercia cuando el contexto lo exige.

La ausencia de ese límite es lo que permite al Coding Trance instalarse y pro-

fundizarse sin resistencia. La presencia de ese límite es lo que distingue a las organizaciones que eventualmente colapsan de las que eventualmente se adaptan. El trance es cómodo. El trance es eficiente. El trance produce indicadores que satisfacen a la junta y tranquilizan al comité ejecutivo. Pero el trance es también una forma de ceguera progresiva que hace invisible su propia existencia. Nadie elige entrar en trance. El trance emerge cuando las condiciones lo permiten. Y las condiciones lo permiten cuando no hay nada externo al sistema que fuerce a la organización a confrontar lo que el sistema no ve.

El loop del poder que se describió al inicio de este libro, el ciclo de autoamplificación que lleva a runaway dynamics, encuentra en el Coding Trance su forma más refinada y más peligrosa. No un loop que se siente como aceleración descontrolada, sino un loop que se siente como control perfecto. No una crisis que obliga a reaccionar, sino una calma que adormece la capacidad de reaccionar. Cuando la junta exija explicaciones sobre la falta de anticipación, la respuesta honesta será: el sistema que usamos para ver estaba diseñado para ver otras cosas, y no teníamos nada fuera de ese sistema que nos obligara a mirar más allá.

La protección contra el Coding Trance no viene de mejor tecnología, ni de personal más capacitado, ni de liderazgo más visionario. Viene de un mecanismo externo, verificable, con autoridad para intervenir antes de que sea demasiado tarde. Pero aquí surge la objeción previsible: ¿por qué externo? ¿No pueden las organizaciones crear sus propios mecanismos de límite? ¿No existen ya comités de revisión, gates de aprobación, instancias de gobernanza diseñadas precisamente para esto?

Existen. Y sistemáticamente fallan. No por falta de diseño ni por deficiencia de las personas que los operan, sino por una asimetría estructural que hace que los límites internos se erosionen inevitablemente. Entender por qué esto ocurre es el paso previo a cualquier solución que pretenda funcionar.