

Servicialo

Un Protocolo Abierto para la Orquestación de
Servicios Profesionales en la Economía de
Agentes de Inteligencia Artificial

Versión 0.6.0 — Marzo 2026
Licencia MIT

Citación APA:

Servicialo. (2026). Servicialo: Un protocolo abierto para la orquestación de servicios profesionales en la economía de agentes de inteligencia artificial (Versión 0.6.0). <https://servicialo.com>

Citación ISO 690:

SERVICIALO. Servicialo: Un protocolo abierto para la orquestación de servicios profesionales en la economía de agentes de inteligencia artificial [en línea]. Versión 0.6.0. Marzo 2026. Disponible en: <https://servicialo.com>

<https://servicialo.com>
<https://github.com/servicialo/mcp-server>

Versión 0.6.0 — Marzo 2026

Citación APA:

Servicialo. (2026). *Servicialo: Un protocolo abierto para la orquestación de servicios profesionales en la economía de agentes de inteligencia artificial (Versión 0.6.0)*. <https://servicialo.com>

Citación ISO 690:

SERVICIALO. *Servicialo: Un protocolo abierto para la orquestación de servicios profesionales en la economía de agentes de inteligencia artificial [en línea]*. Versión 0.6.0. Marzo 2026 [consultado en 2026]. Disponible en: <https://servicialo.com>

Resumen ejecutivo

Los servicios profesionales representan una proporción significativa de la actividad económica global, pero su coordinación permanece fragmentada en silos verticales incompatibles. Un profesional de la salud, un abogado, un docente y un técnico de mantenimiento del hogar enfrentan problemas estructuralmente idénticos — agendar, confirmar, entregar, documentar, cobrar y verificar — pero cada uno utiliza herramientas que modelan estos procesos de formas incompatibles. Esta fragmentación se agrava con la llegada de agentes de inteligencia artificial (IA) que necesitan descubrir, coordinar y liquidar servicios de forma programática, sin vocabulario común ni garantías de interoperabilidad.

Servicialo propone un protocolo abierto, licenciado bajo MIT, que define la capa de orquestación para la entrega de servicios profesionales. El protocolo modela cada servicio a través de 8 dimensiones canónicas — identidad, proveedor, cliente, agenda, ubicación, ciclo de vida, evidencia y cobro — y define un ciclo de vida de 9 estados universales que cualquier servicio debe recorrer desde la solicitud hasta la verificación final. Complementariamente, define la Orden de Servicio como el acuerdo bilateral que agrupa servicios atómicos bajo condiciones comerciales pactadas.

El protocolo incluye flujos de excepción de primera clase, un mecanismo de resolución de disputas con arbitraje algorítmico, un modelo de decisión para agentes de IA con fronteras de autonomía explícitas, y un servidor de Protocolo de Contexto de Modelos (MCP) con 20 herramientas organizadas en 6 fases del ciclo de vida. Servicialo no es un producto: es infraestructura neutral de protocolo que cualquier plataforma puede implementar como nodo soberano.

Palabras clave

Protocolo abierto, servicios profesionales, agentes de inteligencia artificial, ciclo de vida, orquestación, interoperabilidad, Model Context Protocol (MCP), resolución de disputas, prueba de entrega, gobernanza de datos, portabilidad, soberanía de datos.

Índice

1. [El problema](#)
2. [Anatomía del servicio — las 8 dimensiones](#)
3. [El recurso físico como entidad de primera clase](#)
4. [Las dos entidades — Servicio y Orden de Servicio](#)
5. [El ciclo de vida — 9 estados universales](#)
6. [Flujos de excepción](#)
7. [Resolución de disputas — el mecanismo de 80/20](#)
8. [Contrato de servicio pre-acordado](#)
9. [Evidencia por vertical](#)
10. [Principios del estándar](#)
11. [Gobernanza y portabilidad](#)
12. [Cumplimiento regulatorio](#)
13. [Trazabilidad de agentes de inteligencia artificial](#)
14. [Política de versionado](#)
15. [Módulos](#)
16. [Servidor de Protocolo de Contexto de Modelos](#)
17. [Por qué Servicialo](#)
18. [Especificación técnica v0.6.0](#)
19. [Cómo participar](#)

1. El problema

La coordinación de servicios profesionales es un problema que trasciende industrias. Un profesional de la salud que atiende pacientes en consulta, un abogado que asesora a clientes corporativos, un docente particular que enseña a domicilio y un técnico que repara instalaciones comparten una estructura operativa idéntica: alguien solicita, alguien entrega, se acuerda cuándo y dónde, se ejecuta el servicio, se documenta, se cobra y se verifica. Sin embargo, cada vertical ha desarrollado sus propias herramientas, vocabularios y flujos, creando silos incompatibles.

1.1 Fragmentación estructural

Las herramientas existentes modelan fragmentos del problema:

Herramienta	Qué modela	Qué ignora
APIs de agendamiento	Cuándo	Quién paga, qué evidencia, cómo se resuelven excepciones
Pasarelas de pago	Cuánto	El ciclo de vida del servicio, la prueba de entrega
Historias clínicas	Qué datos clínicos	Agendamiento, cobro, verificación del cliente

Herramienta	Qué modela	Qué ignora
CRMs genéricos	Relación comercial	Ciclo de vida operativo, excepciones, evidencia

Ninguna de estas herramientas modela la cadena de valor completa: quién provee, quién recibe, quién paga, cuándo, dónde, qué evidencia prueba que la entrega ocurrió, y qué documentación resultó. El resultado es que la reconciliación operativa — la verificación de que lo acordado fue efectivamente entregado y cobrado correctamente — es un proceso manual, propenso a errores, que consume recursos administrativos desproporcionados.

1.2 El problema de los agentes de IA

La emergencia de agentes de inteligencia artificial capaces de actuar en nombre de personas y organizaciones introduce un segundo problema: estos agentes necesitan un vocabulario común para descubrir servicios disponibles, evaluar proveedores, reservar horarios, verificar entregas y liquidar pagos. Sin un protocolo compartido, cada agente debe aprender las idiosincrasias de cada plataforma, lo que replica a nivel de máquina la misma fragmentación que ya existe a nivel humano.

1.3 La analogía con la infraestructura de internet

La historia de la tecnología muestra que los problemas de coordinación a escala se resuelven con protocolos abiertos, no con plataformas centralizadas:

Protocolo	Qué estandariza	Implementaciones
HTTP	Transferencia de documentos	Servidores web diversos
SMTP	Envío de correo electrónico	Proveedores de correo diversos
SQL	Consulta de datos relacionales	Motores de base de datos diversos
DNS	Resolución de nombres	Servidores de nombres diversos

Servicialo aplica el mismo patrón a los servicios profesionales: define el estándar (el protocolo), y cualquier plataforma puede implementarlo como un nodo soberano en la red. Ninguna entidad es dueña del protocolo. El valor colectivo de la red crece con cada nodo que se conecta.

2. Anatomía del servicio — las 8 dimensiones

Todo servicio profesional — sin importar la vertical — puede describirse a través de 8 dimensiones. Estas son los campos mínimos requeridos para que un agente de IA pueda comprender y coordinar un servicio de extremo a extremo.

#	Dimensión	Descripción	Ejemplos
1	Identidad (Qué)	La actividad o resultado que se entrega	Sesión de rehabilitación, consulta legal, tutoría, reparación eléctrica
2	Proveedor (Quién entrega)	El profesional o entidad que ejecuta el servicio	Profesional de la salud certificado, abogado colegiado, docente, técnico

#	Dimensión	Descripción	Ejemplos
3	Cliente (Quién recibe)	El beneficiario del servicio, con pagador separado explícitamente	Paciente, empresa, estudiante, propietario
4	Agenda (Cuándo)	La ventana temporal acordada	Fecha, hora de inicio, duración esperada
5	Ubicación (Dónde)	El lugar físico o virtual de entrega, incluyendo el recurso físico cuando aplica	Consultorio, domicilio, videollamada, sala de reuniones
6	Ciclo de vida (Estados)	Posición actual en los 9 estados universales	Solicitado → Agendado → ... → Verificado
7	Evidencia (Prueba)	Cómo se demuestra que el servicio ocurrió	GPS, firma digital, fotos, documentos, duración registrada
8	Cobro (Liquidación)	Liquidación financiera con estado independiente del ciclo	Monto, moneda, pagador, estado de pago, documento tributario

2.1 Identidad (Qué)

La primera dimensión define el servicio como un producto: tipo, vertical, nombre legible, duración esperada y prerequisites. Un servicio que no está productizado — que carece de nombre, precio y duración definidos — es invisible para un agente de IA, porque no hay información suficiente para descubrirlo ni coordinarlo.

2.2 Proveedor (Quién entrega)

El proveedor tiene un identificador, credenciales verificables, un puntaje de confianza calculado a partir de su historial operativo, y pertenece a una organización. El puntaje de confianza no es una calificación subjetiva: se computa a partir de datos operativos verificables — tasa de asistencia, cumplimiento de horarios, tasa de disputas, calidad de documentación.

2.3 Cliente (Quién recibe)

Decisión de diseño: El pagador se separa explícitamente del cliente. En salud, la aseguradora paga. En servicios corporativos, la empresa paga. En educación, el apoderado paga. La mayoría de las APIs de agendamiento ignoran esta distinción, lo que genera modelos de datos que no pueden representar la realidad financiera de los servicios profesionales.

2.4 Agenda (Cuándo)

Tres marcas temporales definen la dimensión temporal: cuándo se solicitó el servicio, cuándo se agendó, y cuánto dura. La separación entre solicitud y agendamiento permite medir el tiempo de respuesta operativo.

2.5 Ubicación (Dónde)

La modalidad de entrega puede ser presencial, virtual o a domicilio. Cuando el servicio requiere un espacio físico específico — un consultorio, una sala, un equipamiento — la ubicación referencia una entidad de Recurso con su propio calendario de disponibilidad (ver [Sección 3](#)).

2.6 Ciclo de vida (Estados)

La posición del servicio en su ciclo de vida de 9 estados, con historial completo de transiciones y excepciones registradas. Ver [Sección 5](#).

2.7 Evidencia (Prueba de entrega)

Cómo se demuestra que el servicio ocurrió. Incluye registro de entrada y salida con marca temporal, duración real calculada automáticamente, y evidencia específica según la vertical: GPS, firmas digitales, fotografías, documentos clínicos, minutas de reunión, registros de asistencia.

2.8 Cobro (Liquidación financiera)

El cobro tiene su propio estado, independiente del ciclo de vida. Un servicio puede estar en estado Cobrado en el ciclo de vida mientras su facturación aún está en estado **facturado** (esperando reembolso de aseguradora, por ejemplo).

Decisión de diseño: Los estados **cobrado** y **pagado** se separan explícitamente. **Cobrado** significa que el monto fue debitado del saldo del cliente o agregado a su deuda — ocurre 1:1 con cada sesión completada. **Pagado** significa que el efectivo fue recibido, y puede haber ocurrido antes (cuando el cliente compró un paquete prepago) o después (reembolso de aseguradora). En servicios profesionales donde el modelo dominante es el paquete prepago, confundir cobro con pago pierde información crítica sobre el flujo de caja.

3. El recurso físico como entidad de primera clase

Un Recurso es una entidad física — un consultorio, una silla dental, una cancha deportiva, un box de atención — que un servicio puede requerir para su entrega. Es opcional: las sesiones virtuales, las visitas a domicilio y los servicios entregados en la ubicación del cliente no tienen Recurso. Pero cuando un Recurso existe, es una entidad de primera clase con su propia identidad, calendario de disponibilidad y restricciones.

3.1 Esquema del recurso

Campo	Tipo	Descripción
id	texto	Identificador único
nombre	texto	Nombre legible: "Sala A", "Box 3", "Cancha 1"
tipo	texto	Categoría: sala, box, sillón, equipamiento
capacidad	entero	Máximo de sesiones simultáneas. Por defecto: 1
buffer_minutos	entero	Tiempo de reseteo entre usos. Por defecto: 0
equipamiento	texto[]	Equipamiento disponible
activo	booleano	Si está disponible para reservas
reglas	objeto	Restricciones de lógica de negocio

3.2 Por qué el recurso es una dimensión separada

Un Recurso tiene su propio calendario de disponibilidad, independiente del proveedor y del cliente. Puede estar bloqueado por razones ajenas a cualquier sesión — mantenimiento, limpieza profunda, reserva institucional, calibración de equipos. Tratarlo como un campo de texto en la ubicación — como hacen la mayoría de los sistemas de agendamiento — colapsa dos conceptos distintos y genera conflictos que solo se manifiestan en producción: el proveedor está disponible, el cliente está disponible, pero la sala está en mantenimiento.

Una implementación correcta modela el Recurso como una entidad que participa en la intersección de disponibilidad junto con el proveedor y el cliente. Al agendar una sesión, el sistema debe verificar disponibilidad tripartita:

```
proveedor_libre cliente_libre recurso_libre
```

3.3 El problema del buffer

El buffer no es tiempo del proveedor ni tiempo del cliente — es tiempo del recurso. Un profesional puede atender al siguiente cliente inmediatamente. El consultorio necesita 15 minutos de sanitización. Si el buffer vive en la agenda del proveedor, el modelo es incorrecto — y el error se multiplica cuando el mismo patrón aparece en odontología (esterilización de instrumentos), clases grupales (limpieza de sala) o espacios de coworking (reseteo de puesto).

`buffer_minutos` es un campo de primera clase en el Recurso — no está enterrado dentro de `reglas` — porque el agendador lo necesita para aritmética. La ocupación efectiva de un recurso es `duración_sesión + buffer_minutos`. Esto es una operación matemática, no una regla de negocio.

3.4 Capacidad y sesiones grupales

Cuando `capacidad > 1`, el Recurso puede alojar múltiples sesiones simultáneas siempre que el total de clientes no exceda la capacidad:

```
clientes_actuales + nuevos_clientes ≤ recurso.capacidad
```

Esto no es un caso especial — es el comportamiento general. `capacidad = 1` es simplemente el caso de sesión individual. Un estudio de yoga con `capacidad = 20` y un sillón dental con `capacidad = 1` usan exactamente la misma lógica de agendamiento; solo el número difiere.

4. Las dos entidades — Servicio y Orden de Servicio

El protocolo define exactamente dos entidades centrales que modelan la totalidad de la relación entre proveedor y cliente.

4.1 El Servicio (unidad atómica)

El Servicio es la unidad atómica de entrega. Cada instancia de un servicio es un evento concreto: una sesión de rehabilitación el martes a las 10:00, una consulta legal el viernes a las 15:00, una clase particular el lunes a las 17:00. Un Servicio tiene las 8 dimensiones descritas en la sección anterior y recorre el ciclo de vida de 9 estados de forma independiente.

4.2 La Orden de Servicio (acuerdo bilateral)

La Orden de Servicio es un acuerdo bilateral que agrupa uno o más servicios bajo condiciones comerciales definidas. Tres ejes la definen completamente:

Eje	Descripción	Ejemplo
Alcance	Qué servicios están autorizados, cuántos, de qué tipo	12 sesiones de rehabilitación, 40 horas de consultoría
Precio	Cómo se valora la entrega	Monto fijo, tiempo y materiales, tarifa por nivel, mixto
Calendario de pago	Cuándo se mueve el dinero	Anticipo, por hitos, periódico, contra entrega, personalizado

La Orden de Servicio tiene su propio ciclo de vida:

borrador → propuesta → negociando → activa → pausada → completada
cancelada

Decisión de diseño: Los estados **borrador** y **propuesta** convierten a la Orden de Servicio en el objeto central para cotizaciones. Una cotización ES una Orden de Servicio en estado pre-activo. No existe un objeto "cotización" separado.

4.3 El libro mayor computado

La Orden de Servicio contiene un libro mayor (**ledger**) que es enteramente computado — nunca se ingresa manualmente. A medida que los Servicios atómicos transitan al estado **verificado**, el sistema actualiza automáticamente los campos del libro mayor:

Campo	Descripción
servicios_verificados	Conteo de servicios en estado verificado
horas_consumidas	Horas totales de servicios verificados
monto_consumido	Valor consumido a las tarifas pactadas
monto_facturado	Total facturado a la fecha
monto_cobrado	Total de pagos recibidos
monto_restante	Alcance autorizado no consumido

Esto significa que la Orden de Servicio siempre refleja el estado real de la entrega sin ningún paso de reconciliación. El cierre de mes del administrador se convierte en una revisión de excepciones, no en una reconstrucción desde cero.

4.4 Por qué son dos objetos separados

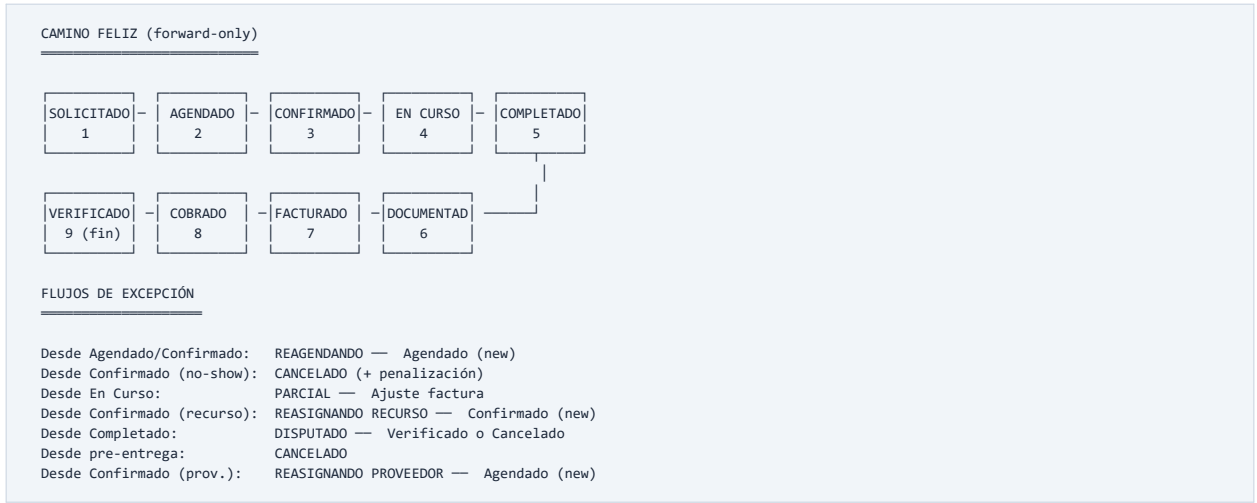
Principio: El acuerdo es separado de la entrega. La Orden de Servicio define lo acordado. Los servicios atómicos definen lo entregado. Confundirlos — como hacen la mayoría de los sistemas de agendamiento y facturación — crea un problema fundamental de integridad de datos: no se puede determinar si una disputa es sobre los términos del acuerdo o sobre la calidad de la entrega.

5. El ciclo de vida — 9 estados universales

Todo servicio profesional — sin importar la vertical — recorre el mismo ciclo de vida. Los 9 estados son el mínimo requerido para que un agente de IA pueda verificar con certeza que un servicio fue solicitado, entregado, documentado y liquidado.

Solicitado → Agendado → Confirmado → En Curso → Completado → Documentado → Facturado → Cobrado → Verificado

Diagrama de máquina de estados



#	Estado	Descripción	Disparador
1	Solicitado	El cliente o su agente define qué necesita, cuándo y dónde	El cliente envía la solicitud
2	Agendado	Se asigna hora, proveedor, ubicación y recurso (si aplica)	El sistema cruza disponibilidad
3	Confirmado	Ambas partes reconocen el compromiso. Prerrequisitos verificados	Proveedor y cliente confirman
4	En Curso	Registro de entrada detectado. El servicio está siendo entregado	Detección de check-in
5	Completado	El proveedor marca la entrega como completa. Evidencia capturada	El proveedor confirma
6	Documentado	Registro formal generado: ficha, reporte, minuta — según la vertical	Documentación archivada
7	Facturado	Documento tributario emitido	Boleta o factura generada
8	Cobrado	Pago recibido y confirmado	Pago acreditado
9	Verificado	El cliente confirma que el servicio ocurrió y fue cobrado correctamente, o se auto-verifica tras la ventana de silencio	Confirmación o silencio

5.1 ¿Por qué 9 estados?

Menos estados pierden información crítica:

- Sin separar Completado de Documentado, no se puede distinguir "el proveedor dice que ocurrió" de "la evidencia está en el registro".
- Sin separar Facturado de Cobrado, no se puede saber si el pago fue efectivamente recibido.
- Sin separar Cobrado de Verificado, no se puede saber si el cliente aceptó el resultado.

Más estados agregan fricción innecesaria. 9 es el conjunto mínimo viable para que un agente de IA verifique la cadena completa del servicio con certeza.

5.2 ¿Por qué Verificado es el último estado?

La verificación es el cierre del ciclo, no un paso intermedio. En la práctica:

1. El proveedor entrega (Completado)
2. El proveedor documenta (Documentado)
3. Se emite el documento tributario (Facturado)
4. Se recibe el pago (Cobrado)
5. El cliente verifica — o la ventana de verificación expira y se auto-cierra (Verificado)

El cliente no puede verificar de forma significativa hasta que el servicio ha sido documentado, facturado y cobrado. Necesita el cuadro completo — la documentación, la factura, la confirmación de pago — antes de poder confirmar o disputar. Una verificación que ocurre antes de la documentación es prematura.

5.3 Compromiso tripartito en estado Agendado

Cuando una sesión tiene un Recurso asignado, el estado Agendado implica que tres entidades están simultáneamente comprometidas: el proveedor, el cliente y el recurso físico. Este es un compromiso más fuerte que una cita bilateral — liberar cualquiera de los tres afecta a los otros dos. La implementación debe tratar el compromiso tripartito como atómico.

5.4 Estados intermedios

Las implementaciones pueden agregar estados entre los 9 universales para adecuarse a su realidad operativa. Por ejemplo, un paso de asignación entre Solicitado y Agendado, o una revisión de calidad entre Documentado y Facturado. Los 9 estados universales son el mínimo — no el máximo.

5.5 Transiciones de estado

Los estados son estrictamente ordenados. No se pueden saltar estados universales (por ejemplo, pasar de Agendado a Documentado). Cada transición registra:

- Desde qué estado y hacia qué estado
- Cuándo ocurrió (marca temporal)
- Quién la disparó (cliente, proveedor, sistema o agente de IA)
- Cómo se disparó (**auto**, **manual** o **agent**)
- Metadatos específicos del contexto

5.6 Regla de nómina

Las implementaciones que calculan la compensación de proveedores deben leer únicamente sesiones en estado **Cobrado**. Las sesiones que aún no han alcanzado ese estado no son hechos consumados y no deben contar para la nómina. Esto elimina el modo de falla común donde los proveedores registran sesiones retroactivamente al cierre de mes para inflar su compensación.

6. Flujos de excepción

Un protocolo robusto no solo define el camino feliz. Define qué ocurre cuando las cosas fallan. Estos son flujos de primera clase, no casos de borde. Estadísticamente, las excepciones ocurren en el 15–30% de todas las citas de servicio.

6.1 Inasistencia del cliente

Disparador: El cliente no llega dentro del período de gracia.

Confirmado → Cancelado (inasistencia)

- Se aplica penalización según la política de la organización
- Se libera el horario del proveedor para reasignación
- Se incrementa el contador de inasistencias del cliente (sistema de strikes)
- El proveedor es compensado según la política

6.2 Inasistencia del proveedor

Disparador: El proveedor no llega o cancela a último momento.

Confirmado → Reasignando → Agendado (nuevo proveedor)

- El sistema busca automáticamente un proveedor de reemplazo
- El cliente es notificado del cambio
- El proveedor original es marcado

6.3 Cancelación

Disparador: Cualquiera de las partes cancela antes del servicio.

Cualquier estado pre-entrega → Cancelado

- Se aplica la política de cancelación según el tiempo restante
- Reembolso total si está fuera de la ventana de penalización
- Reembolso parcial o nulo dentro de la ventana

6.4 Disputa de calidad

Disparador: El cliente disputa la calidad de un servicio completado dentro de la ventana de disputa.

Completado → Disputado

- El cobro se congela automáticamente
- Se solicita evidencia adicional de ambas partes
- Administración o arbitraje resuelve
- Se resuelve hacia: Cobrado → Verificado (proveedor gana) o Cancelado (cliente gana, saldo restaurado)

6.5 Reagendamiento

Disparador: Cualquiera de las partes necesita cambiar el horario.

Agendado/Confirmado → Reagendando → Agendado (nuevo horario)

- Busca horario compatible para ambas partes (y recurso, si aplica)
- Mantiene al mismo proveedor cuando es posible
- La política de reagendamiento puede aplicar cargos

6.6 Entrega parcial

Disparador: El servicio no puede completarse en su totalidad.

En Curso → Parcial

- Se documenta lo que fue entregado
- Se ajusta la factura proporcionalmente
- Se agenda continuación si es necesario

6.7 Conflicto de recurso

Disparador: El recurso asignado deja de estar disponible después de la confirmación — por mantenimiento, emergencia, falla de equipamiento o error de agendamiento.

Confirmado → Reasignando recurso → Confirmado (nuevo recurso) | Reagendando (sin alternativa)

- El sistema busca un recurso alternativo que satisfaga los mismos requisitos (capacidad, equipamiento) dentro del mismo horario
- **Si encuentra alternativa:** la sesión se reasigna. El proveedor siempre es notificado. El cliente se notifica solo si el cambio es material (diferente ubicación, diferentes características)
- **Si no encuentra alternativa:** la excepción escala a un flujo de Reagendamiento

Decisión de diseño: El conflicto de recurso es una excepción distinta de la inasistencia del proveedor porque la lógica de resolución es fundamentalmente diferente. Un reemplazo de proveedor cambia el quién; un reemplazo de recurso cambia el dónde. La mayoría de los clientes se preocupan profundamente por qué profesional los atiende y menos por en qué sala ocurre. Esta asimetría exige que las reglas de notificación y los umbrales de escalamiento se modelen por separado.

7. Resolución de disputas — el mecanismo de 80/20

La resolución de disputas en servicios profesionales sigue un patrón predecible: aproximadamente el 80% de los casos pueden resolverse de forma algorítmica, evaluando la evidencia disponible contra las reglas contractuales. El 20% restante requiere juicio humano a través de arbitraje por pares.

7.1 Flujo de disputa

Paso	Etap	Descripción	Actor
1	Apertura	Cualquier parte abre una disputa dentro del plazo definido. Se congela el cobro automáticamente	Cliente, proveedor o agente
2	Revisión de evidencia	Se solicita evidencia adicional de ambas partes. Se evalúa contra el contrato	Sistema o administración
3	Resolución	Si el proveedor gana: Cobrado → Verificado. Si el cliente gana: Cancelado con saldo restaurado	Administración o arbitraje

7.2 Resolución algorítmica (~80% de los casos)

El sistema evalúa la evidencia disponible contra las reglas definidas en el contrato de servicio:

- ¿Existen registros de entrada y salida con metadatos válidos?
- ¿La documentación está completa y firmada según lo requerido?
- ¿La duración real coincide con la duración pactada dentro de un margen aceptable?
- ¿La lista de verificación acordada fue completada?

Cuando la evidencia satisface los criterios contractuales de forma unívoca, la disputa se resuelve automáticamente sin intervención humana. Esto reduce costos operativos y tiempos de resolución.

7.3 Arbitraje por pares (~20% de los casos)

Cuando la evidencia es ambigua o insuficiente para una resolución algorítmica, la disputa se escala a arbitraje por pares del mismo vertical. Profesionales con experiencia verificada en la misma disciplina evalúan el caso. La configuración del arbitraje se define en el contrato de servicio:

- Número de árbitros según el monto en disputa
- Monto máximo que puede disputarse sin escalamiento externo
- Ventana temporal para la resolución

8. Contrato de servicio pre-acordado

Antes de que cualquier servicio pueda ser reservado, el contrato de servicio define las reglas operativas que ambas partes aceptan. Este contrato no es un documento legal extenso — es un conjunto estructurado de campos que un agente de IA puede leer y aplicar de forma determinista.

Campo	Descripción	Ejemplo
evidencia_requerida	Qué evidencia debe registrarse para considerar el servicio entregado	Registro de entrada + registro de salida + documentación firmada
plazo_disputa	Ventana de tiempo para abrir una disputa después de Completado	48 horas
política_cancelación	Reglas de penalización por cancelación según tiempo restante	0% si >24h, 50% si 2–24h, 100% si <2h
política_inasistencia	Qué ocurre si una parte no se presenta	Cliente: cobra 100%. Proveedor: reasignación + penalidad
arbitraje	Configuración del arbitraje por pares si aplica	1 árbitro si monto < \$50, 3 si ≥ \$50
monto_máximo_disputa	Monto máximo que puede disputarse sin escalamiento externo	Equivalente a \$500 USD

El contrato de servicio **debe** ser consultado antes de cualquier acción de reserva o transición de estado. Esto garantiza que tanto humanos como agentes de IA operan bajo las mismas reglas explícitas.

9. Evidencia por vertical

La prueba de entrega no es genérica — cada vertical tiene requisitos de evidencia específicos que reflejan sus obligaciones regulatorias, prácticas profesionales y expectativas del cliente. El protocolo define evidencia requerida por vertical, y cada implementación puede extenderla según sus necesidades.

9.1 Salud

Tipo de evidencia	Descripción	Automática
Registro de entrada	Marca temporal GPS del proveedor al llegar	Sí
Registro de salida	Marca temporal GPS del proveedor al salir	Sí
Ficha clínica firmada	Registro clínico firmado por profesional y paciente	No
Adherencia al plan	Lista de verificación del plan de tratamiento ejecutado	No

Regla de resolución: Si registros de entrada/salida existen y ficha clínica está firmada por ambas partes, servicio entregado. Si falta ficha o firma, escalar.

9.2 Hogar

Tipo de evidencia	Descripción	Automática
Foto antes	Foto del estado inicial con marca temporal y GPS	No
Foto después	Foto del resultado final con marca temporal y GPS	No
Lista de verificación	Tareas acordadas marcadas como completadas	No
Firma del cliente	Firma digital del cliente confirmando recepción	No

Regla de resolución: Si fotos antes/después existen con metadatos válidos y lista de verificación completa, servicio entregado. Si falta firma del cliente, escalar.

9.3 Legal

Tipo de evidencia	Descripción	Automática
Minuta de reunión	Registro de lo discutido y acordado	No
Entrega de documentos	Confirmación de entrega de documentos generados	No
Registro de horas	Horas facturables con descripción de actividades	No

Regla de resolución: Si minuta existe y horas registradas están dentro del rango acordado, servicio entregado. Si horas exceden lo acordado sin justificación, escalar.

9.4 Educación

Tipo de evidencia	Descripción	Automática
Registro de asistencia	Confirmación de presencia del alumno y profesor	Sí
Entrega de material	Material o tareas entregadas al alumno	No
Registro de evaluación	Evaluación o retroalimentación de la sesión	No

Regla de resolución: Si asistencia registrada y material entregado, servicio entregado. Si falta evaluación y contrato la requiere, escalar.

9.5 Consultoría

Tipo de evidencia	Descripción	Automática
Entregable aprobado	Documento o artefacto entregado con aprobación formal del cliente	No
Acta de comité	Registro de sesión de comité o reunión de seguimiento con asistentes y acuerdos	No
Registro de horas	Horas facturables con descripción de actividades por categoría	No
Avance de hito	Porcentaje de avance contra el hito definido en la Orden de Servicio	No

Regla de resolución: Si entregable aprobado por el cliente y horas registradas están dentro del presupuesto autorizado, servicio entregado. Si horas exceden el presupuesto sin autorización de cambio de alcance, escalar.

9.6 Resumen comparativo

Vertical	Evidencia automatizable	Evidencia manual	Criterio clave de resolución
Salud	GPS entrada/salida	Ficha clínica, adherencia	Ficha firmada por ambas partes
Hogar	—	Fotos, checklist, firma	Fotos con metadatos válidos + checklist completa
Legal	—	Minuta, documentos, horas	Horas dentro del rango acordado
Educación	Asistencia	Material, evaluación	Asistencia + material entregado

Vertical	Evidencia automatizable	Evidencia manual	Criterio clave de resolución
Consultoría	—	Entregable, acta, horas, hito	Entregable aprobado + horas dentro de presupuesto

9.7 Extensibilidad

Cada vertical puede definir tipos de evidencia adicionales. El protocolo define la estructura (tipo, marca temporal de captura, datos específicos), no el catálogo cerrado. Una implementación para servicios de traducción puede agregar "archivo traducido con control de calidad"; una para servicios de arquitectura puede agregar "planos aprobados por el mandante". La clave es que cada tipo de evidencia tiene una regla de resolución que un agente de IA puede evaluar de forma determinista.

10. Principios del estándar

Seis principios guían el diseño del protocolo. No son aspiraciones — son restricciones de diseño que informan cada decisión técnica.

Principio 1: Todo servicio tiene un ciclo.

No importa si es una sesión de rehabilitación o una auditoría financiera. Los 9 estados del ciclo de vida son universales para cualquier servicio profesional. Las particularidades de cada estado varían por vertical, pero la secuencia es invariante.

Principio 2: La entrega debe ser verificable.

Si no se puede probar que el servicio ocurrió, no ocurrió. El estándar define qué constituye evidencia válida para que tanto humanos como agentes de IA puedan confiar en ella. Los tipos de evidencia incluyen: GPS, duración registrada, documentos firmados, fotografías y confirmación del cliente.

Principio 3: El pagador no siempre es el cliente.

En salud paga la aseguradora. En servicios corporativos la empresa. En educación el apoderado. El protocolo separa explícitamente al beneficiario, al solicitante y al pagador como entidades independientes.

Principio 4: Las excepciones son la regla.

Inasistencias, cancelaciones, reagendamientos, disputas — no son casos de borde. Ocurren en el 15–30% de todas las citas de servicio. Un servicio bien diseñado define qué pasa cuando algo falla.

Principio 5: Un servicio es un producto.

Tiene nombre, precio, duración, requisitos y resultado esperado. Definido así, tanto un humano como un agente de IA pueden descubrirlo y coordinarlo. Los servicios que no están productizados son invisibles para los agentes.

Principio 6: El acuerdo es separado de la entrega.

La Orden de Servicio define lo acordado. Los servicios atómicos definen lo entregado. Son objetos distintos con ciclos de vida distintos. El libro mayor computado en la Orden de Servicio es el puente entre ambos.

11. Gobernanza y portabilidad

11.1 El cliente como dueño de su información

El protocolo establece que los datos operativos pertenecen a las partes que los generan — fundamentalmente, al cliente. Una persona que recibe servicios profesionales a lo largo de su vida genera un historial que incluye sesiones completadas, documentación clínica o profesional, pagos realizados y disputas resueltas. Este historial no debería estar cautivo en una plataforma.

11.2 Portabilidad de datos

Cualquier implementación del protocolo debe permitir que el cliente exporte su historial en un formato interoperable definido por el estándar. Esto incluye:

- Historial de servicios con sus 8 dimensiones
- Documentación generada (fichas, reportes, minutas)
- Historial de pagos y facturación
- Excepciones y disputas con sus resoluciones

11.3 Soberanía de los nodos

Cada organización que implementa el protocolo opera como un nodo soberano. Esto significa:

- **Propiedad de datos:** La organización retiene soberanía completa sobre sus datos operativos
- **Autonomía operativa:** Cada nodo decide qué verticales atender, qué precios cobrar, qué políticas de cancelación aplicar
- **Interoperabilidad voluntaria:** Los nodos se conectan al protocolo por voluntad propia y pueden desconectarse sin perder sus datos

11.4 Inteligencia de red como bien común

Cuando múltiples nodos contribuyen datos agregados y anónimos a la red, la inteligencia colectiva resultante — patrones de demanda, benchmarks de precios, métricas de eficiencia operativa — es un bien común del protocolo. Ninguna implementación puede capturar, revender o monopolizar esta inteligencia.

El modelo de contribuir-para-acceder

El acceso a la inteligencia de red es proporcional a la contribución. Este modelo garantiza beneficio simétrico e incentiva la participación honesta.

Qué contribuye cada nodo (snapshot mensual, anónimo y agregado):

Categoría	Métricas contribuidas	Ejemplo
Volumen	Cantidad de servicios por estado, por vertical	340 servicios verificados en el vertical salud

Categoría	Métricas contribuidas	Ejemplo
Eficiencia temporal	Tiempo promedio entre estados (solicitud→agendado, agendado→confirmado, etc.)	Mediana de 2.3 horas entre Solicitado y Agendado
Excepciones	Tasas de inasistencia, cancelación, reagendamiento y disputa	8.2% de inasistencia de clientes, 1.1% de disputas
Precios	Distribución de precios por tipo de servicio (percentiles, no valores individuales)	Percentil 25/50/75 de sesiones de rehabilitación: \$25K / \$35K / \$50K
Utilización de recursos	Tasa de ocupación de recursos físicos por tipo	Consultorios al 72% de capacidad promedio
Conversión	Tasa de conversión de propuestas a órdenes activas	43% de órdenes propuestas llegan a estado activo

Qué recibe cada nodo a cambio:

Nivel de contribución	Benchmarks recibidos
Básico (volumen + excepciones)	Promedios generales de su vertical y región
Estándar (+ precios + eficiencia)	Percentiles por vertical, región y escala comparable
Completo (todas las categorías)	Dashboard comparativo con posición relativa en cada métrica, tendencias temporales y alertas de desviación

Reglas de privacidad del modelo:

- Todo dato es agregado y anónimo — nunca se comparten datos de clientes, proveedores o sesiones individuales
- Tamaño mínimo de segmento: 5 organizaciones por celda (vertical × región) para prevenir re-identificación
- Los nodos retienen soberanía completa sobre sus datos operativos — la contribución a la red es una copia agregada, no un acceso a los datos fuente
- La extensión de telemetría se activa cuando el ecosistema alcanza 10+ organizaciones con datos consistentes

Gobernanza del dato de red:

Los datos contribuidos a la red son gobernados por el protocolo, no por ninguna implementación individual. Las decisiones sobre política de datos siguen el proceso de RFC del protocolo. Principios inviolables:

1. Ningún nodo puede acceder a datos desagregados de otro nodo
2. El dato de red no puede venderse, sublicenciarse ni monetizarse directamente
3. Cualquier nodo puede dejar de contribuir en cualquier momento, perdiendo acceso proporcionalmente
4. Los benchmarks generados pertenecen al protocolo como bien común

12. Cumplimiento regulatorio

El protocolo está diseñado para facilitar el cumplimiento de las principales regulaciones de protección de datos personales vigentes en las jurisdicciones donde los servicios profesionales operan.

12.1 Ley 19.628 (Chile) — Protección de Datos Personales

Requisito	Cómo el protocolo lo facilita
Consentimiento informado	El contrato de servicio define explícitamente qué datos se recopilan y para qué
Finalidad del tratamiento	Cada campo del esquema tiene una descripción que documenta su propósito
Derecho de acceso y rectificación	La portabilidad de datos permite al cliente exportar y revisar su información
Derecho de cancelación	El protocolo soporta la eliminación de datos personales preservando registros agregados anónimos

12.2 RGPD (Unión Europea) — Reglamento General de Protección de Datos

Requisito	Cómo el protocolo lo facilita
Base legal del tratamiento	El contrato de servicio pre-acordado establece la base contractual
Minimización de datos	Las 8 dimensiones definen los campos mínimos necesarios — no se recopilan datos superfluos
Portabilidad (Art. 20)	Exportación en formato estructurado, de uso común y lectura mecánica
Derecho al olvido (Art. 17)	Separación entre datos operativos (eliminables) y datos agregados anónimos (retenibles)
Privacidad por diseño (Art. 25)	La arquitectura del protocolo integra privacidad desde la definición del esquema

12.3 LGPD (Brasil) — Lei Geral de Proteção de Dados

Requisito	Cómo el protocolo lo facilita
Bases legales (Art. 7)	Contrato de servicio como base para el tratamiento de datos
Derechos del titular (Art. 18)	Acceso, corrección, portabilidad y eliminación soportados por el protocolo
Transferencia internacional	El esquema es agnóstico a jurisdicción; las reglas de transferencia las define cada nodo

12.4 CCPA (California, EE.UU.) — California Consumer Privacy Act

Requisito	Cómo el protocolo lo facilita
Derecho a saber (§1798.100)	El esquema documenta cada campo recopilado y su propósito
Derecho a eliminar (§1798.105)	Soportado a nivel de protocolo con preservación de agregados anónimos
Derecho a no vender (§1798.120)	La inteligencia de red usa exclusivamente datos anónimos y agregados — nunca datos individuales

12.5 Principio general

El protocolo no reemplaza la asesoría legal ni garantiza cumplimiento por sí solo. Provee la infraestructura técnica para que cada implementación pueda cumplir con las regulaciones aplicables en su jurisdicción. La responsabilidad del cumplimiento recae en cada nodo implementador.

13. Trazabilidad de agentes de inteligencia artificial

Los agentes de IA son ciudadanos de primera clase en el protocolo, pero no todos los estados son iguales desde la perspectiva de autonomía. Algunas transiciones son deterministas y seguras para que un agente las ejecute solo. Otras involucran ambigüedad, dinero real o consecuencias irreversibles que requieren confirmación humana.

13.1 Modelo de decisión para agentes

Transición	Agente puede actuar solo	Requiere confirmación humana
Solicitado → Agendado	Sí — el sistema cruza disponibilidad	—
Agendado → Confirmado	Sí — si ambas partes confirmaron	Si la confirmación es ambigua
Confirmado → En Curso	Sí — al detectar check-in	—
En Curso → Completado	—	Sí — el proveedor debe marcar
Completado → Documentado	Sí — si evidencia auto-capturada	Si requiere documentación manual
Documentado → Facturado	Sí — si reglas de cobro definidas	Si requiere cálculo manual
Facturado → Cobrado	Sí — al confirmar pago	—
Cobrado → Verificado	Sí — al confirmar cliente o expirar ventana	—
Cualquier estado → Cancelado	—	Sí — siempre requiere acción humana
Cualquier estado → Disputado	—	Sí — el cliente debe iniciar

13.2 Reglas de trazabilidad

Cada transición de estado registra el campo **by** (quién disparó) y **method** (cómo se disparó). Cuando un agente de IA ejecuta una transición, el registro incluye:

- **Identificador del agente:** Un ID único que identifica al agente específico
- **Método:** **agent** — distinguiéndolo de **auto** (sistema) y **manual** (humano)
- **En nombre de:** Si el agente actúa en representación de un cliente o proveedor, la referencia a la persona representada

13.3 La regla de ambigüedad

Cuando una transición cae en la categoría de "requiere confirmación humana", el agente debe pausar y presentar la ambigüedad a un humano antes de proceder. El agente nunca debe resolver ambigüedad por suposición. Debe presentar la información disponible y la decisión requerida, luego esperar.

13.4 La regla de irreversibilidad

Cualquier transición que mueve dinero, genera un documento legal o cierra una Orden de Servicio es irreversible por defecto. Los agentes deben tratar estas transiciones como que requieren confirmación humana explícita, independientemente de cuán determinista parezca el disparador.

13.5 Auditoría de agentes

El historial completo de transiciones de un servicio permite auditar retrospectivamente qué acciones fueron tomadas por agentes de IA, cuáles por humanos y cuáles por el sistema. Esto es fundamental para:

- **Responsabilidad:** Determinar quién tomó cada decisión en la cadena
- **Cumplimiento regulatorio:** Demostrar que las decisiones automatizadas tienen trazabilidad completa
- **Mejora continua:** Identificar patrones donde los agentes necesitan ajustes en sus fronteras de autonomía

14. Política de versionado

El protocolo sigue versionado semántico:

Nivel	Significado	Ejemplo
Parche (0.6.x)	Clarificaciones, correcciones tipográficas, ejemplos adicionales	0.6.1
Menor (0.x.0)	Nuevos campos opcionales, nuevos flujos de excepción, extensiones	0.7.0
Mayor (x.0.0)	Cambios incompatibles en campos requeridos o modelo de estados	1.0.0

14.1 Independencia de versiones

La versión del protocolo es independiente de la versión del servidor MCP ([@servicialo/mcp-server](#)). Ambas se rastrean por separado. Una actualización del servidor MCP puede incluir mejoras de implementación sin cambiar la versión del protocolo.

14.2 Compatibilidad hacia atrás

Los cambios menores son siempre aditivos — agregan campos opcionales o extensiones que las implementaciones existentes pueden ignorar sin romperse. Los cambios mayores se comunican con anticipación suficiente para que las implementaciones puedan migrar.

14.3 Proceso de cambio

Los cambios al protocolo se proponen mediante issues públicos en el repositorio del proyecto, se discuten abiertamente, y se incorporan tras consenso de la comunidad de implementadores.

15. Módulos

El protocolo se organiza en módulos que permiten adopción incremental. Una implementación puede comenzar con el módulo núcleo y agregar módulos adicionales según sus necesidades.

15.1 Servicialo Core (Estable)

Todo lo necesario para modelar un servicio profesional de principio a fin.

Audiencia: Cualquier plataforma donde dos partes toman un compromiso de entrega y necesitan una cuenta verificable de lo que ocurrió.

Incluye:

- Ciclo de vida completo (9 estados universales)
- Las 8 dimensiones del servicio
- Recurso físico como entidad de primera clase
- Órdenes de servicio (acuerdo comercial + libro mayor computado)
- Flujos de excepción (cancelación, inasistencia, reagendamiento, disputa, entrega parcial, conflicto de recurso)
- Prueba de entrega con evidencia configurable por vertical
- Contrato de servicio pre-acordado
- Servidor MCP para agentes de IA (20 herramientas en 6 fases)

15.2 Servicialo/Finanzas (En diseño)

Distribución de pagos entre las partes involucradas.

Audiencia: Plataformas que intermedian pagos entre clientes y profesionales, o que cobran comisiones e infraestructura.

Incluye:

- Distribución de pagos a tres destinatarios (profesional, organización, infraestructura)
- Tipos de distribución: porcentaje, monto fijo, mixto
- Momentos de liquidación: por sesión, mensual, al cierre
- Concepto de infraestructura (recurso físico como componente del costo)

15.3 Servicialo/Disputas (En diseño)

Resolución formal de disputas con arbitraje algorítmico y por pares.

Audiencia: Plataformas con volumen suficiente para justificar arbitraje estructurado — o donde el monto por servicio hace que las disputas sean económicamente relevantes.

Incluye:

- Flujo de disputa estructurado en 3 pasos
- Resolución algorítmica (~80% de los casos)
- Arbitraje por pares del mismo vertical

- Evidencia válida definida por vertical
- Configuración de umbrales y escalamiento

16. Servidor de Protocolo de Contexto de Modelos

Servicialo expone sus herramientas como un servidor de Model Context Protocol (MCP), habilitando a los agentes de IA para descubrir y coordinar servicios profesionales de forma nativa.

16.1 Modos de operación

Modo	Herramientas disponibles	Requisitos
Descubrimiento	4 herramientas públicas	Sin configuración
Autenticado	20 herramientas completas	API key + ID de organización

16.2 Las 6 fases y 20 herramientas

Fase 1: Descubrir (4 herramientas públicas)

Siempre disponibles, sin autenticación.

Herramienta	Descripción
<code>registry.search</code>	Buscar organizaciones por vertical y ubicación
<code>registry.get_organization</code>	Obtener detalles públicos de una organización
<code>scheduling.check_availability</code>	Verificar horarios disponibles (agendador de 3 variables: proveedor □ cliente □ recurso)
<code>services.list</code>	Listar el catálogo público de servicios de una organización

Fase 2: Entender (2 herramientas)

Comprensión de la estructura del servicio y los términos contractuales antes de comprometerse.

Herramienta	Descripción
<code>service.get</code>	Obtener las 8 dimensiones de un servicio
<code>contract.get</code>	Obtener el contrato de servicio pre-acordado. Debe llamarse antes de cualquier acción de Fase 3+

Fase 3: Comprometer (3 herramientas)

Reserva y confirmación de una sesión de servicio.

Herramienta	Descripción
<code>clients.get_or_create</code>	Encontrar cliente por email/teléfono o crear si es nuevo
<code>scheduling.book</code>	Reservar nueva sesión (estado "Solicitado"). Acepta <code>resource_id</code> opcional para reservar recurso físico
<code>scheduling.confirm</code>	Confirmar sesión reservada (estado "Confirmado"). Ambas partes deben confirmar

Fase 4: Ciclo de vida (4 herramientas)

Gestión de transiciones de estado y flujos de excepción.

Herramienta	Descripción
<code>lifecycle.get_state</code>	Obtener estado actual, transiciones disponibles e historial
<code>lifecycle.transition</code>	Ejecutar transición de estado con evidencia
<code>scheduling.reschedule</code>	Flujo de excepción: reagendar a nuevo horario
<code>scheduling.cancel</code>	Flujo de excepción: cancelar sesión con política contractual

Fase 5: Verificar entrega (3 herramientas)

Registro de prueba de entrega del servicio.

Herramienta	Descripción
<code>delivery.checkin</code>	Registrar check-in con GPS + marca temporal (estado "En Curso")
<code>delivery.checkout</code>	Registrar check-out con GPS + marca temporal (estado "Completado"). Duración auto-calculada
<code>delivery.record_evidence</code>	Registrar evidencia de entrega por vertical: GPS, firma, foto, documento, duración, notas

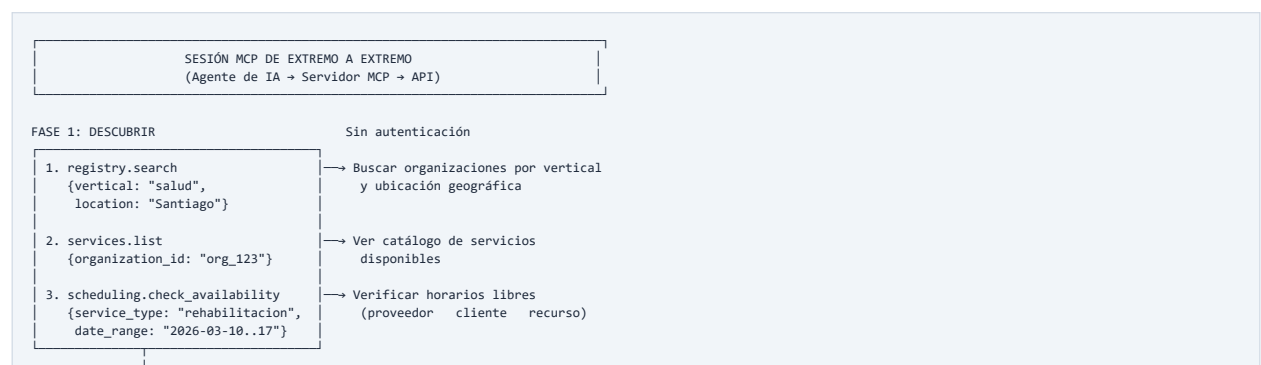
Fase 6: Cerrar (4 herramientas)

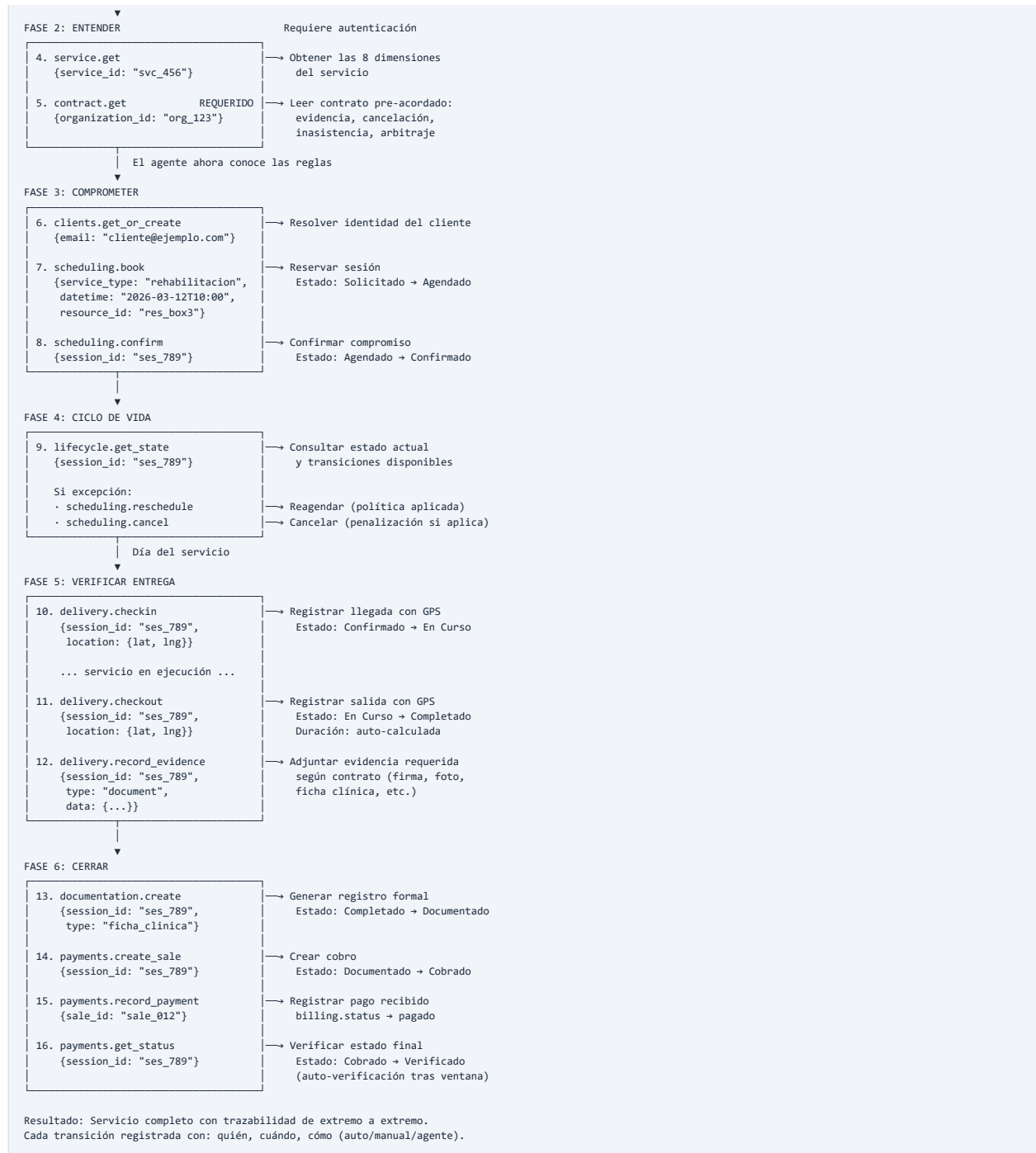
Documentación, facturación y liquidación de pago.

Herramienta	Descripción
<code>documentation.create</code>	Generar registro del servicio: ficha clínica, reporte de inspección, minuta de clase (estado "Documentado")
<code>payments.create_sale</code>	Crear un cobro por servicio documentado (estado "Cobrado")
<code>payments.record_payment</code>	Registrar pago recibido. El pago es independiente del ciclo de vida
<code>payments.get_status</code>	Consultar estado de pago de una venta o historial de saldo del cliente

16.3 Flujo de extremo a extremo — sesión MCP completa

El siguiente diagrama muestra cómo un agente de IA recorre las 6 fases para coordinar un servicio profesional completo, desde el descubrimiento hasta el cierre:





Nota: Este flujo representa el camino feliz. En cualquier punto entre las Fases 3–5, un flujo de excepción (cancelación, reagendamiento, inasistencia, disputa) puede desviar el servicio del camino principal. El agente consulta `lifecycle.get_state` para conocer las transiciones disponibles en cada momento.

16.4 Instalación

```
# Modo descubrimiento (sin credenciales)
npx -y @servicialo/mcp-server

# Modo autenticado (con credenciales)
# Configurar variables: SERVICIALO_API_KEY, SERVICIALO_ORG_ID
```

16.4 Paquete

- **npm:** `@servicialo/mcp-server`
- **Repositorio:** <https://github.com/servicialo/mcp-server>

17. Por qué Servicialo

17.1 La lógica de red

El valor de un protocolo de servicios profesionales crece exponencialmente con cada nodo que lo implementa, siguiendo la ley de Metcalfe. Un nodo aislado obtiene los beneficios del ciclo de vida estructurado y la prueba de entrega verificable. Dos nodos pueden intercambiar referencias de clientes con historial verificable. Diez nodos generan benchmarks estadísticamente significativos por vertical y región. Cien nodos crean una red de inteligencia colectiva que beneficia a todos los participantes.

La analogía es directa: en redes de navegación vehicular, cada conductor contribuye datos GPS en tiempo real. Ninguna contribución individual tiene valor por sí sola. Pero el agregado — patrones de tráfico, rutas óptimas, detección de incidentes — beneficia a cada conductor proporcionalmente. La red se vuelve más inteligente a medida que crece.

En servicios profesionales, la inteligencia de red incluye:

- **Patrones de demanda:** Qué servicios se solicitan más, cuándo, dónde
- **Benchmarks de precios:** Distribuciones de precios por vertical, región y nivel de experiencia
- **Métricas de eficiencia:** Tasas de inasistencia, tiempos de confirmación, tasas de disputa
- **Señales de calidad:** Correlaciones entre evidencia de entrega y satisfacción del cliente

17.2 La oportunidad regulatoria

Las regulaciones de protección de datos en múltiples jurisdicciones convergen en tres requisitos: portabilidad, consentimiento informado y minimización de datos. Un protocolo abierto que integra estos principios desde su diseño no solo facilita el cumplimiento — posiciona a sus implementadores en ventaja ante futuras regulaciones que amplíen estos requisitos.

Regulación de inteligencia artificial

La regulación de sistemas de IA está avanzando rápidamente en múltiples jurisdicciones, y los servicios profesionales se encuentran en el epicentro de esta convergencia:

Jurisdicción	Regulación	Requisito relevante	Cómo Servicialo lo anticipa
Unión Europea	AI Act (Reglamento de IA)	Transparencia y trazabilidad para sistemas de IA de alto riesgo; registro de decisiones automatizadas	Cada transición registra <code>method: agent</code> con ID del agente y referencia a la persona representada
Unión Europea	AI Act — Art. 14	Supervisión humana de sistemas de IA	El modelo de decisión para agentes define fronteras explícitas de autonomía con escalamiento obligatorio a humanos

Jurisdicción	Regulación	Requisito relevante	Cómo Servicialo lo anticipa
Unión Europea	AI Act — Art. 13	Transparencia hacia usuarios afectados	El campo by en cada transición identifica si la acción fue tomada por humano, sistema o agente
América Latina	Marcos regulatorios de IA en desarrollo (Chile, Brasil, Colombia, México)	Responsabilidad algorítmica y derechos de los afectados	La auditoría retrospectiva del historial de transiciones permite determinar responsabilidad en cada decisión
Global	Tendencia hacia "derecho a explicación"	Los usuarios deben poder entender por qué se tomó una decisión automatizada	Los metadatos de cada transición documentan el contexto y los criterios aplicados

Convergencia regulatoria de datos y de IA

La intersección de regulaciones de protección de datos (RGPD, LGPD, CCPA) con regulaciones de IA crea un escenario donde los servicios profesionales coordinados por agentes necesitan cumplir simultáneamente con ambos marcos. Un protocolo que integra trazabilidad de agentes y protección de datos desde su arquitectura reduce significativamente la carga de cumplimiento para cada implementador — en lugar de resolver estos requisitos caso a caso, la infraestructura del protocolo los resuelve una vez para todos los nodos.

Los implementadores que adopten el protocolo antes de que estas regulaciones se consoliden tendrán la ventaja de operar con una base técnica que ya satisface los requisitos emergentes, en lugar de tener que adaptar retrospectivamente sistemas que no fueron diseñados para la trazabilidad.

17.3 El modelo económico

Servicialo como protocolo es libre y gratuito (licencia MIT). No cobra por uso, no cobra por transacción, no cobra por volumen. El valor económico se captura en la capa de implementación — cada nodo monetiza según su modelo de negocio particular: suscripciones, comisiones, licencias o servicios de consultoría.

Este modelo replica la dinámica probada de los protocolos exitosos: HTTP es libre, pero los servidores web, los CDN y las plataformas de hosting generan valor sobre él. SQL es libre, pero los motores de base de datos comerciales y los servicios gestionados capturan el valor en la capa de implementación.

18. Especificación técnica v0.6.0

```
#
# SERVICIALO v0.6.0
# Especificación técnica completa
#

servicio:
  id: texto                # Identificador único
  orden_de_servicio_id: texto # Opcional – referencia a Orden padre
  tipo: texto              # Categoría del servicio
  vertical: texto          # salud | legal | hogar | educación | consultoría | ...
  nombre: texto            # Nombre legible
  duración_minutos: entero  # Duración esperada
  requisitos: texto[]       # Prerrequisitos

proveedor:
  id: texto                # Dimensión 2 – Quién entrega
  credenciales: texto[]     # Certificaciones requeridas
  puntaje_confianza: número # 0-100, calculado por historial
  organización_id: texto    # Organización padre

cliente:
  id: texto                # Dimensión 3 – Quién recibe
  pagador_id: texto        # Puede diferir del cliente

agenda:
  solicitado_en: fecha_hora # Dimensión 4 – Cuándo
```

```

    agendado_para: fecha_hora
    duración_esperada: minutos

ubicación:
    # Dimensión 5 – Dónde
    tipo: presencial | virtual | domicilio
    dirección: texto
    sala: texto
    # Etiqueta legible (coexiste con recurso_id)
    coordenadas:
        lat: número
        lng: número
    recurso_id: texto
    # Opcional – referencia a entidad Recurso

recurso:
    # Dimensión 5b – Recurso físico (opcional)
    id: texto
    nombre: texto
    # "Box 3", "Sala A", "Sillón 2"
    tipo: texto
    # box | sala | sillón | equipamiento
    capacidad: entero
    # Máximo de clientes simultáneos. Default: 1
    buffer_minutos: entero
    # Tiempo de reseteo entre usos. Default: 0
    equipamiento: texto[]
    ubicación: texto
    # Descripción física (piso, ala, dirección)
    activo: booleano
    # Si está disponible para reservas
    reglas: objeto
    # Restricciones extensibles de lógica de negocio

disponibilidad_recurso:
    # Calendario recurrente del recurso
    recurso_id: texto
    día_semana: entero
    # 0 = Domingo, 6 = Sábado
    hora_inicio: texto
    # "HH:mm"
    hora_fin: texto
    # "HH:mm"
    disponible: booleano
    # true = disponible, false = bloqueado
    zona_horaria: texto
    # IANA timezone
    orden_bloque: entero
    # Secuencia dentro del día

ciclo_de_vida:
    # Dimensión 6 – Estados
    estado_actual: enum
    # Los 9 estados universales
    - solicitado
    - agendado
    - confirmado
    - en_curso
    - completado
    - documentado
    - facturado
    - cobrado
    - verificado
    estados_excepción:
    # Estados fuera del camino feliz
    - cancelado
    - disputado
    - reasignando
    - reagendando
    - parcial
    transiciones: transición[]
    # Historial de cambios
    excepciones: excepción[]
    # Inasistencias, disputas, etc.

prueba_de_entrega:
    # Dimensión 7 – Evidencia
    entrada: fecha_hora
    salida: fecha_hora
    duración_real: minutos
    # Auto-calculada
    evidencia: evidencia[]
    # GPS, firma, fotos, documentos

cobro:
    # Dimensión 8 – Liquidación financiera
    monto:
        valor: número
        moneda: texto
        # ISO 4217 (CLP, USD, MXN, BRL, EUR, ...)
    pagador: referencia
    # Puede diferir del cliente
    estado: pendiente | cobrado | facturado | pagado | disputado
    cobrado_en: fecha_hora
    # 1:1 con estado Cobrado del ciclo
    pago_id: referencia
    # Pago vinculado (puede ser paquete prepago)
    documento_tributario: ref
    # Boleta/factura si se emitió

# -----
# ORDEN DE SERVICIO
# Acuerdo bilateral que agrupa servicios
# -----

orden_de_servicio:
    id: texto
    # Identificador único
    organización_id: texto
    # Organización emisora
    cliente_id: texto
    # Beneficiario
    pagador_id: texto
    # Quién paga

alcance:
    # Qué servicios están autorizados
    descripción: texto
    # Definición legible del alcance
    tipos_servicio: texto[]
    # Tipos autorizados
    límite_cantidad: entero | nulo
    # Máximo de servicios. null = ilimitado
    límite_horas: número | nulo
    # Máximo de horas. null = no aplica
    condición_expiración: texto
    # Condición de finalización legible

plazo:
    # Duración y renovación
    tipo: permanente | anual | mensual | por_hito | por_evento
    inicia_en: fecha_hora
    termina_en: fecha_hora | nulo
    # null si permanente
    auto_renueva: booleano
    # Default: false
    días_aviso_renovación: entero | nulo

precio:
    # Cómo se valora la entrega
    modelo: fijo | tiempo_materiales | tarifa | mixto
    moneda: texto
    # ISO 4217
    monto_fijo: número
    # Requerido si modelo = fijo
    tarifa:
        - nivel: texto
        # junior | senior | partner
        tarifa_facturable: número
        # Por hora, al cliente
        tarifa_costo: número
        # Por hora, costo interno

calendario_pago:
    # Cuándo se mueve el dinero
    tipo: anticipo | hito | periódico | contra_entrega | personalizado

```

```

cuotas:
  - disparador: texto          # "contrato_firmado" | "hito_1_aprobado"
    monto: número | nulo
    porcentaje: número | nulo
    días_vencimiento: entero

libro_mayor:                  # Computado – nunca ingresado manualmente
  servicios_verificados: entero
  horas_consumidas: número
  monto_consumido: número
  monto_facturado: número
  monto_cobrado: número
  monto_restante: número

ciclo_de_vida:
  estado_actual: borrador | propuesta | negociando | activa | pausada | completada | cancelada
  transiciones: transición[]

ids_servicio: texto[]        # Referencias a servicios atómicos

# -----
# TIPOS DE SOPORTE
# -----

transición:
  desde: texto | nulo        # null para estado inicial
  hacia: texto
  en: fecha_hora
  por: texto                 # ID de cliente, proveedor, sistema o agente
  método: auto | manual | agente
  metadatos: objeto

excepción:
  tipo: inasistencia | cancelación | disputa | reagendamiento | parcial | conflicto_recurso
  en: fecha_hora
  iniciado_por: texto
  resolución: texto
  resuelto_en: fecha_hora

evidencia:
  tipo: gps | firma | foto | documento | duración | notas
  capturado_en: fecha_hora
  datos: objeto             # Payload específico del tipo

actor:
  tipo: cliente | proveedor | organización | agente
  id: texto
  en_nombre_de:             # Opcional – cuando un agente actúa en representación
    tipo: cliente | proveedor
    id: texto

contrato_servicio:
  evidencia_requerida: texto[] # Tipos de evidencia necesarios para entrega válida
  plazo_disputa: texto        # Ventana para disputar (e.g., "48 horas")
  politica_cancelación: objeto # Reglas de penalización por tiempo restante
  politica_inasistencia: objeto # Consecuencias de no-show
  arbitraje: objeto           # Configuración de arbitraje por pares
  monto_máximo_disputa: número # Umbral de escalamiento externo

```

19. Cómo participar

Servicialo es un protocolo abierto bajo licencia MIT. Existen tres formas de participar:

19.1 Implementadores

Organizaciones que construyen plataformas de servicios profesionales pueden implementar el protocolo como nodo soberano.

Requisitos mínimos para ser listado como implementación compatible:

1. Modelar servicios usando las 8 dimensiones
2. Implementar los 9 estados del ciclo de vida
3. Manejar al menos 3 flujos de excepción
4. Exponer una API que el servidor MCP pueda conectar
5. (Opcional) Modelar Órdenes de Servicio
6. (Opcional) Contribuir a la inteligencia de red

19.2 Proveedores de servicios

Profesionales y organizaciones que entregan servicios pueden adoptar el protocolo a través de cualquier implementación compatible. Los beneficios incluyen:

- Historial profesional portátil entre plataformas
- Puntaje de confianza basado en datos verificables, no en reseñas subjetivas
- Acceso a benchmarks de la industria para optimizar su operación

19.3 Desarrolladores de agentes de IA

Equipos que construyen agentes de IA pueden integrar el servidor MCP para que sus agentes descubran, coordinen y liquiden servicios profesionales.

Inicio rápido:

```
npx -y @servicialo/mcp-server
```

Con 4 herramientas públicas disponibles sin autenticación, un agente puede inmediatamente buscar organizaciones, explorar catálogos de servicios y verificar disponibilidad.

19.4 Contribuciones al protocolo

- **Diseño del protocolo:** Proponer cambios a dimensiones, estados o principios vía issues públicos
- **Extensiones:** Diseñar módulos adicionales (Finanzas, Disputas, Telemetría)
- **Traducciones:** La especificación se mantiene en español e inglés
- **Documentación:** Mejorar ejemplos, guías de implementación y casos de uso

Repositorio: <https://github.com/servicialo/mcp-server>

Sitio web: <https://servicialo.com>

Licencia

MIT — Usar, implementar, extender. La atribución se agradece pero no es obligatoria.

Servicialo es un protocolo abierto mantenido por su comunidad de implementadores. Ninguna empresa es dueña del protocolo. La inteligencia colectiva de la red es un bien común.

<https://servicialo.com>