GUÍA PRÁCTICA

Cómo sistematizar subagentes y automatizar procesos desde ChatGPT

Aplicación a gestión de costos de construcción con enfoque PMI + Lean Construction

Incluye: arquitectura, roles de subagentes, flujo de automatización, implementación por etapas, controles, pruebas y conexión con Excel / base de datos / Lovable.

Versión de trabajo · Octubre 2026

Objetivo del documento

Convertir una idea de “varios agentes especializados” en un sistema operativo concreto: un orquestador que recibe la solicitud, distribuye tareas, consulta información compartida, valida resultados y entrega una salida trazable. La guía distingue qué se puede hacer manualmente en ChatGPT y qué requiere una aplicación con API para automatización real.

Idea clave: ChatGPT en una conversación no convierte por sí solo varios GPTs separados en un equipo que se ejecuta de manera autónoma. Para delegación y ejecución automática entre agentes, se necesita una capa de orquestación, normalmente construida con la API y el Agents SDK, más las herramientas y datos que cada agente debe usar.

1. Elegir el nivel de automatización

Antes de construir, define cuánto querés automatizar y qué herramientas estás dispuesta a usar.

Nivel	Cómo funciona	Cuándo elegirlo	Límite principal
1. Asistido en ChatGPT	Un chat principal usa prompts maestros. La persona copia o comparte resultados entre especialistas.	Probar el método sin programar ni invertir en integración.	No hay ejecución autónoma entre chats ni sincronización automática garantizada.
2. Semi-automatizado	ChatGPT se conecta a herramientas o APIs autorizadas; un flujo externo mueve datos y dispara tareas.	Automatizar partes repetitivas y conservar aprobación humana.	Necesita configurar integraciones, permisos y reglas.
3. Multiagente real	Una aplicación usa OpenAI API / Agents SDK para coordinar especialistas, ejecutar herramientas y validar resultados.	Operación sistemática, trazable y escalable.	Requiere desarrollo, pruebas, almacenamiento y monitoreo.

Recomendación para este sistema

Construir por etapas: comenzar con un piloto asistido en ChatGPT; formalizar los contratos de datos y los criterios de validación; luego desarrollar el orquestador con API / Agents SDK. No automatizar pagos, órdenes de compra ni cambios de presupuesto sin una aprobación explícita de una persona responsable.

Importante para la cuenta

La disponibilidad de creación de GPTs personalizados depende del plan y del espacio de trabajo. La documentación oficial indica que la creación y publicación de nuevos GPTs no está disponible en cuentas personales Free, Go, Plus y Pro; los GPTs existentes pueden seguir utilizándose. Esta guía no depende de que puedas crear GPTs personalizados: la automatización multiagente se puede construir mediante una aplicación conectada a la API.

2. Diseñar la arquitectura del sistema

Arquitectura recomendada para Gestión de Costos PMI / Lean Construction

Componente	Responsabilidad	Entrada / salida
Orquestador / CEO IA	Interpreta la solicitud, decide qué subagentes intervienen, asigna tareas y consolida la respuesta.	Entrada del usuario → plan de trabajo → informe consolidado.
Planificación IA	Revisa WBS/EDT, actividades, secuencia, hitos, restricciones y avance.	Cronograma, restricciones, fechas, cantidades planificadas.
Costos IA	Analiza costos unitarios, consumos, precios, desvíos y supuestos.	APU, cotizaciones, costos reales y variaciones.
Procura IA	Compara proveedores, lead times, condiciones, riesgos de abastecimiento.	Solicitudes de cotización, comparativa y recomendación.
Presupuesto IA	Integra costos directos/indirectos, contingencias, impuestos y versiones.	Presupuesto base, revisión y variación.
Flujo de Caja IA	Proyecta egresos/ingresos por período y analiza necesidades de fondos.	Cash flow, escenarios y alertas.
Riesgos IA	Identifica riesgos, causas, probabilidad, impacto, respuestas y responsables.	Registro de riesgos y exposición económica.
Controller IA	Contrasta resultados, detecta inconsistencias y exige evidencia.	Validaciones, observaciones y estado de aprobación.
Premortem / Postmortem IA	Anticipa fallas antes de ejecutar y captura aprendizajes después.	Acciones preventivas y lecciones aprendidas.
Capa de datos	Conserva la fuente única de información, versiones, permisos y trazabilidad.	Base de datos / archivos / registro de cambios.

Flujo operativo

Usuario → Orquestador → Subagentes requeridos → Controller → (si hay errores, devolución al responsable) → aprobación humana cuando corresponda → actualización de datos / Excel / dashboard → reporte final.

3. Definir contratos de trabajo para cada subagente

No alcanza con darles nombres. Cada subagente necesita una misión acotada, datos de entrada definidos, un formato de salida y condiciones para declarar que no puede continuar.

Plantilla de instrucciones para cada subagente

```
ROL: [nombre del subagente]
OBJETIVO: [resultado concreto que debe producir]
ALCANCE: [qué hace y qué no hace]
ENTRADAS OBLIGATORIAS: [datos y documentos requeridos]
REGLAS:
- No inventar precios, cantidades, fechas, impuestos ni cotizaciones.
- Identificar fuente, fecha, moneda, unidad y versión de cada dato.
- Separar hechos, supuestos, estimaciones y recomendaciones.
- Señalar datos faltantes y detener cálculos críticos cuando corresponda.
SALIDA ESTRUCTURADA:
1. Resultado / resumen
2. Datos utilizados y fuentes
3. Cálculos y fórmulas
4. Supuestos y exclusiones
5. Alertas / riesgos
6. Decisiones requeridas
7. Estado: APTO / OBSERVADO / BLOQUEADO
CRITERIO DE ACEPTACIÓN: [cómo se verifica que la salida es correcta]
ENTREGA A: [orquestador u otro subagente]
```

Reglas de coordinación

- **Una única fuente de verdad:** presupuesto, cantidades, costos reales y cronograma deben tener identificadores y versiones comunes.
- **Sin datos inventados:** si falta una cotización, marcar “pendiente de cotización”; no completar con un precio ficticio.
- **Formato común:** los agentes devuelven campos estructurados, no solo texto libre.
- **Control de cambios:** cada modificación registra quién/qué la originó, fecha, versión anterior, versión nueva y motivo.
- **Escalamiento:** los cambios de alcance, presupuesto base, compromisos contractuales o pagos se elevan a aprobación humana.

4. Paso a paso para un piloto dentro de ChatGPT

Esta fase permite probar el método antes de desarrollar software. Es una automatización asistida, no una red autónoma de subagentes.

Paso 1 — Crear el proyecto de trabajo

Abrí un proyecto o una conversación dedicada al sistema. Subí solo los documentos vigentes: alcance, WBS, cronograma, presupuesto, APU, cotizaciones, registro de riesgos y criterios de control. Si los archivos tienen versiones distintas, identificá cuál es la válida.

Paso 2 — Crear el prompt maestro del orquestador

Indicá que debe entender la solicitud, clasificarla, elegir los especialistas necesarios, enumerar datos faltantes, definir el orden de ejecución y consolidar el resultado. Exigí que cite el archivo o registro usado y que nunca invente información.

Paso 3 — Preparar prompts de subagentes

Redactá un prompt por especialidad usando la plantilla de la sección 3. Cada uno debe tener un alcance estrecho y devolver siempre la misma estructura.

Paso 4 — Ejecutar un caso de prueba pequeño

Ejemplo: revisar una variación del costo de un rubro. El orquestador solicita al agente de costos el análisis, al de procura la evidencia de cotizaciones, al de presupuesto el impacto en la línea base y al controller la verificación.

Paso 5 — Transferir resultados con trazabilidad

Si trabajás con chats separados, copiá el resultado estructurado al chat orquestador y conservá el identificador del caso. No asumas que un chat puede leer automáticamente el contenido de otro.

Paso 6 — Validar antes de emitir

Pedí al controller revisar sumas, unidades, monedas, duplicados, fechas, supuestos, coherencia con la línea base y evidencia. Las observaciones deben volver al subagente responsable.

Paso 7 — Aprobar y guardar

Una persona responsable valida las decisiones que afecten costo, plazo, contratación o flujo de fondos. Después se actualiza la fuente de verdad y se guarda el informe final con fecha y versión.

Paso 8 — Medir el piloto

Registrá tiempo de preparación, porcentaje de datos faltantes detectados, errores encontrados, cantidad de retrabajos y porcentaje de salidas aprobadas sin corrección.

Prompt maestro inicial (copiar y adaptar)

```
Actuá como ORQUESTADOR del Sistema de Gestión de Costos de Construcción PMI + Lean Construction.
Para cada solicitud:
1. Identificá el resultado solicitado, el proyecto, la versión de datos y la decisión que se necesita.
2. Indicá qué subagentes deben intervenir y en qué orden.
3. Verificá que existan los datos mínimos. Si faltan, preguntá o marcá BLOQUEADO; no inventes.
4. Entregá tareas delimitadas a cada especialista, con entradas y formato de salida definidos.
5. Consolidá los resultados y pedí una revisión independiente del Controller.
6. Separá datos comprobados, supuestos, estimaciones y recomendaciones.
7. No autorices por tu cuenta pagos, órdenes de compra ni cambios de línea base.
8. Finalizá con: resumen ejecutivo, alertas, decisiones pendientes, fuentes/versiones y estado APTO /
OBSERVADO / BLOQUEADO.
```

5. Automatización real: API + Agents SDK

Para que los especialistas se invoquen automáticamente, compartan contexto de forma controlada y ejecuten herramientas, se necesita una aplicación de orquestación. OpenAI ofrece el Agents SDK para construir agentes, usar herramientas, coordinar especialistas, aplicar guardrails y observar las ejecuciones. La documentación distingue el patrón de gerente que llama agentes como herramientas y el patrón de handoff en el que un especialista toma el control de la conversación.

Paso 1 — Definir el caso de uso mínimo

Elegí un flujo con bajo riesgo y retorno medible: por ejemplo, análisis de variaciones de costos y generación de alertas, sin emitir órdenes de compra ni efectuar pagos.

Paso 2 — Preparar el entorno técnico

Un desarrollador configura un proyecto en Python o TypeScript, crea el acceso a la API de OpenAI y guarda las claves en un gestor de secretos o variables seguras del servidor. Nunca pongas claves API en un prompt, archivo compartido o interfaz pública.

Paso 3 — Implementar el orquestador

El orquestador recibe el pedido, lo clasifica, valida campos mínimos y decide qué agentes llamar. Para mantener una respuesta consolidada, suele convenir el patrón “agentes como herramientas”: el gerente conserva la responsabilidad de la salida final.

Paso 4 — Implementar los especialistas

Definí cada agente con instrucciones específicas y herramientas limitadas. Ejemplos de herramientas: leer presupuesto vigente, consultar cotizaciones, calcular variaciones, registrar riesgos o generar un borrador de informe.

Paso 5 — Crear contratos de datos

Definí un esquema común para proyecto_id, paquete_trabajo, código_rubro, moneda, unidad, cantidad, costo_base, costo_real, fecha_fuente, versión y estado_validación. Validá los campos antes de guardar o pasar información al siguiente agente.

Paso 6 — Conectar la fuente de verdad

Usá una base de datos o repositorio controlado. Excel puede ser entrada/salida inicial, pero evitá que varias automatizaciones editen simultáneamente el mismo archivo sin control de versiones.

Paso 7 — Conectar herramientas externas

Implementá funciones/API para consultar datos, leer archivos, escribir registros autorizados y generar reportes. Si conectás un GPT o interfaz a una API externa, definí autenticación, permisos y esquema de la API.

Paso 8 — Añadir guardrails y aprobación humana

Bloqueá acciones de alto impacto hasta que una persona apruebe. Validá límites de presupuesto, monedas, duplicados, permisos, integridad de datos y consistencia entre agentes.

Paso 9 — Probar con casos conocidos

Usá ejemplos con resultados esperados: una variación correcta, una cotización vencida, un dato faltante, moneda incompatible, duplicado y un cambio de alcance. Compará la salida con el resultado validado por un experto.

Paso 10 — Observar y mejorar

Guardá trazas de qué agente actuó, qué herramienta usó, qué datos consultó, qué validaciones pasaron y dónde falló. Ajustá prompts y lógica con pruebas repetibles antes de pasar a producción.

6. Diseñar el flujo de automatización

Ejemplo: detectar y analizar una desviación de costo

Etapa	Acción automática	Regla de control
1. Disparador	Se carga un costo real nuevo o se actualiza una cotización.	El registro incluye proyecto, fecha, moneda, unidad, fuente y versión.
2. Orquestador	Clasifica la variación y asigna tareas.	No procesa si faltan campos obligatorios.
3. Costos IA	Calcula diferencia absoluta y porcentual contra la base.	Compara magnitudes equivalentes y la misma moneda o conversión documentada.
4. Procura IA	Comprueba cotización, vigencia, proveedor, plazo y condiciones.	No considera cotización vencida como oferta vigente.
5. Presupuesto IA	Calcula impacto en rubro, paquete y presupuesto total.	No cambia la línea base sin aprobación.
6. Riesgos IA	Evalúa exposición, causa probable y respuesta.	Distingue riesgo identificado de hecho confirmado.
7. Controller	Revisa fórmulas, trazabilidad, consistencia y supuestos.	Si falla una regla crítica, devuelve OBSERVADO o BLOQUEADO.
8. Aprobación	Envía una recomendación al responsable.	Ninguna compra ni cambio contractual se ejecuta automáticamente sin permiso.
9. Salida	Actualiza el registro y genera informe / alerta / dashboard.	Conserva fecha, versión, autor y evidencia.

Disparadores posibles

- Carga o actualización de un archivo de presupuesto, costos reales o cotizaciones.
- Llegada de una fecha de revisión semanal o mensual.
- Cambio de un dato crítico en la base de datos.
- Solicitud manual desde una interfaz interna.
- Alerta por umbral: desviación porcentual, cotización próxima a vencer, restricción crítica o déficit proyectado de caja.

7. Excel, base de datos y dashboard Lovable

La interfaz no debe ser la fuente de verdad por sí sola. Conviene separar la capa de datos, la lógica de negocio, la orquestación y la visualización.

Capa	Función recomendada	Decisión de diseño
Excel	Modelo inicial, importación controlada, análisis detallado y entregable descargable.	Usar tablas con identificadores, nombres de campos estables y validaciones. No depender de celdas sin nombre ni fórmulas opacas.
Base de datos	Registro persistente de proyectos, rubros, costos, cotizaciones, riesgos, versiones y aprobaciones.	Una entidad por tipo de dato; claves únicas y registro de cambios.
API / orquestador	Coordina agentes y aplica reglas; conecta con los datos.	El modelo propone y analiza; las funciones deterministas calculan y guardan.
Lovable	Interfaz visual: filtros, semáforos, gráficos, detalle de rubros y alertas.	Conectar a una API/backend y base de datos; nunca exponer claves secretas en el frontend.
Control de acceso	Define quién consulta, modifica, aprueba o administra.	Permisos mínimos necesarios y aprobación humana para acciones críticas.

Módulos sugeridos del dashboard

- Resumen ejecutivo: presupuesto base, costo comprometido, costo real, estimación al cierre y alertas.
- Control de costos: variaciones por rubro, paquete de trabajo, proveedor y período.
- Procura: estado de cotizaciones, comparativa, lead time y compras pendientes.
- Flujo de caja: egresos previstos/reales, saldo proyectado y brechas de financiación.
- Riesgos: exposición económica, responsables, respuestas y vencimientos.
- Controller: datos incompletos, cálculos rechazados, decisiones pendientes y trazabilidad.
- Premortem / postmortem: riesgos anticipados, causas reales y lecciones reutilizables.

Regla técnica

Las operaciones matemáticas críticas (sumas, porcentajes, conversión monetaria, reglas de tolerancia y validaciones) deben ejecutarse con código o fórmulas verificables, no confiarse únicamente a texto generado por el modelo.

8. Control, seguridad y calidad

Control	Qué verificar	Qué hacer si falla
Calidad de datos	Campos obligatorios, duplicados, unidades, fechas y moneda.	Detener el cálculo y pedir corrección.
Trazabilidad	Fuente, fecha, versión, responsable y evidencia.	Marcar no verificable; no elevar a dato confirmado.
Límites económicos	Umbrales de desviación, compromisos y saldo de caja.	Crear alerta y escalar a responsable.
Aprobación	Compra, pago, cambio de alcance o presupuesto.	Requerir autorización humana registrada.
Seguridad	Claves API, credenciales, permisos y datos sensibles.	Revocar credenciales expuestas; limitar permisos.
Observabilidad	Agente invocado, herramientas, duración, error y resultado.	Registrar, diagnosticar y volver a ejecutar solo con reglas seguras.
Evaluación	Casos de prueba y métricas de precisión.	No desplegar hasta alcanzar criterios acordados.

Métricas para monitorear

- Porcentaje de tareas terminadas sin intervención.
- Porcentaje de salidas aprobadas sin correcciones.
- Errores críticos por cada 100 ejecuciones.
- Porcentaje de datos sin fuente o con fecha vencida.
- Tiempo medio por análisis y costo de ejecución.
- Número de bloqueos correctamente detectados.
- Tiempo entre detectar una desviación y comunicarla.

9. Plan de implementación recomendado

Fase	Entregable	Criterio para avanzar
1. Diseño	Mapa de procesos, roles, reglas, datos y aprobaciones.	El flujo y los responsables están acordados.
2. Piloto manual	Prompts de orquestador, costos, procura, presupuesto y controller.	El equipo puede ejecutar un caso y revisar sus resultados.
3. Estandarización	Esquemas de entrada/salida, catálogos, identificadores y reglas de calidad.	Los agentes producen salidas comparables.
4. MVP API	Orquestador y 2–3 agentes conectados a datos de prueba.	Pasa casos conocidos y no ejecuta acciones no autorizadas.
5. Integración	Base de datos, Excel y dashboard.	La actualización y la trazabilidad funcionan de punta a punta.
6. Automatización programada	Tareas periódicas y alertas por eventos.	Hay monitoreo, recuperación ante fallas y aprobación humana.
7. Producción	Documentación, responsables, soporte y evaluación continua.	Se cumplen criterios de calidad, seguridad y operación.

Checklist de inicio

- Definir el primer caso de uso y sus límites.
- Elegir el responsable de negocio y el responsable técnico.
- Reunir documentos y datos vigentes.
- Crear los prompts de cada rol con formato de salida común.
- Definir reglas de bloqueo y aprobaciones.
- Preparar 10–20 casos de prueba, incluyendo errores y datos faltantes.
- Medir el piloto y decidir si se justifica la integración por API.

10. Recursos oficiales y alcance de la guía

Documentación oficial consultada

OpenAI Agents SDK — guía oficial
https://developers.openai.com/api/docs/guides/agents/sdk
Construcción de agentes con herramientas, contexto y coordinación.

OpenAI Agents — opciones de runtime
https://developers.openai.com/api/docs/guides/agents
Comparación de Agents API, Agents SDK y Responses API.

OpenAI — Orchestration and handoffs
https://developers.openai.com/api/docs/guides/agents/orchestration
Diferencia entre agentes como herramientas y handoffs.

OpenAI Help Center — GPTs en ChatGPT
https://help.openai.com/en/articles/8554407-create-a-custom-gpt
Disponibilidad y funcionamiento de GPTs personalizados.

Qué incluye y qué no incluye

Esta guía aporta una arquitectura y un procedimiento de implementación. No crea ni conecta por sí misma los agentes, no configura credenciales, no construye el backend y no garantiza la disponibilidad de una función en una cuenta específica. La integración real requiere implementar y probar las conexiones, permisos, lógica de datos y controles correspondientes.

Recomendación final

No empieces automatizando todo el sistema. Empezá por un flujo acotado de análisis de variaciones de costos; estandarizá entradas y salidas; incorporá el Controller como control independiente; y recién después conectá procura, presupuesto, flujo de caja, riesgos y dashboard. La automatización confiable depende tanto de los datos, reglas y controles como del modelo de IA.