Guía de desarrollador · quote-developer

quote-developer: guía técnica

Cómo genera el agente el PDF de cotización de alcance y llena la plantilla XLSX del equipo Campus, qué celdas puede tocar y cómo se verifica que ambos cuadren.

3 nov 2026 | Guía de desarrollador | Claude Code | 16 min de lectura

Descripción general

El **Agente Desarrollador de Cotizaciones** (`quote-developer`) convierte los alcances de un proyecto en dos entregables que deben contar exactamente el mismo alcance: un **PDF de cotización de alcance** con el desglose por módulos y precios, y el **XLSX de cotización** del equipo Campus, que corre el modelo de costos con las estimaciones de esfuerzo por especialidad.

Esta guía explica cómo está construido: la jerarquía de módulos, submódulos y funcionalidades que comparten el PDF y el Excel, qué celdas de la plantilla puede llenar y cuáles son intocables, cómo se estiman los días, cómo se diseña el PDF y cómo se comprueba que todo cuadre antes de entregar. Está pensada para quien mantiene el agente.

2	3	10	0,5
entregables que deben contar el mismo alcance: PDF y XLSX	niveles de detalle: módulo, submódulo y funcionalidad	especialidades para estimar días, de las columnas B a K	días es el piso mínimo de cualquier estimación

LO ESENCIAL

Nunca inventa alcance. Toda la información técnica sale de lo que el usuario comparte; ante la duda, pregunta. Y **nunca toca las fórmulas** de la plantilla: solo llena las celdas de datos permitidas y dos celdas obligatorias explícitamente autorizadas.

Qué hace y qué no hace

El agente hace	El agente no hace
Redacta el PDF de alcance por módulos, submódulos y funcionalidades atómicas	Inventar alcance, agrupar funcionalidades para ahorrar espacio ni fragmentar una sola para aparentar detalle
Llena el XLSX con la estimación de días por especialidad y por funcionalidad	Modificar fórmulas, checkboxes ni la tabla de costos de la plantilla
Recalcula el XLSX y escribe en el PDF los precios leídos de él	Calcular o transcribir precios a mano
Pregunta al usuario ante ambigüedad de alcance, esfuerzo o datos	Asumir ni completar huecos con texto genérico
Verifica que PDF y XLSX coincidan antes de entregar	Entregar sin la doble verificación de precios

Cómo corre: Claude Code y `CLAUDE.md`

Corre en **Claude Code** y se gobierna con `CLAUDE.md`, que lee al inicio de cada sesión y se co-evoluciona con el usuario. Las páginas de `wiki/` (`plantilla-xlsx.md`, `diseno-pdf-cotizacion.md`, `perfil-usuario.md`) son **espejos consultables** de partes de `CLAUDE.md`: si alguna difiere, manda `CLAUDE.md`, y los espejos se actualizan junto con él.

Decisiones fundacionales

Tema	Decisión	Por qué
Entregables	PDF de alcance y XLSX de cotización	El PDF documenta el alcance; el XLSX corre el modelo de costos del equipo
Veracidad	Cero invención de alcance	El PDF solo contiene lo que el usuario provee o confirma
Plantilla	Archivo con fórmulas ya montadas por el equipo	El agente solo rellena celdas de datos permitidas

Tema	Decisión	Por qué
Salida	Carpeta propia por cliente	Es autocontenida, igual que en los demás agentes
Idioma	Español	Todo el contenido y la comunicación

Estructura del repositorio

```
quote-developer/
  CLAUDE.md                       # rol, reglas del PDF y del XLSX
  wiki/
    index.md                      # catálogo y resumen de cotizaciones entregadas
    log.md                        # bitácora (solo se añade)
    perfil-usuario.md             # rol y reglas de oro
    marca-campuslands.md          # identidad general + diseño vigente del PDF
    diseno-pdf-cotizacion.md      # espejo del diseño del PDF
    plantilla-xlsx.md             # mapeo de celdas de la plantilla
  cotizaciones/
    <slug-cliente>/               # un cliente
      Cotizacion de Alcance.pdf
      Cotizacion.xlsx
    <slug-cliente>/<slug-proyecto>/  # un cliente con varios proyectos
  recursos/                       # solo lectura
    FullServices - Plantilla Cotizaciones.xlsx  # plantilla maestra: nunca se edita
```

DÓNDE VIVE EL AGENTE Y SUS RECURSOS COMPARTIDOS

El agente está en `agents/claude-agents/quote-developer/`. Los logos, el isotipo, el favicon, el `Brief Fullservice.pdf` y el Brandbook de Campuslands ya no se copian dentro de cada agente: están en `resources/campuslands-resources/`, en la raíz del repositorio (desde la carpeta del agente: `../../../resources/campuslands-resources/`). El logo de cada empresa cliente (por ejemplo, el de Compumax) está en `resources/companies-logos/<empresa>/`.

Reglas de carpetas

- **`wiki/` y `cotizaciones/` son del agente; `recursos/` es solo lectura.**
- Para cada cliente nuevo, el agente **copia** la plantilla maestra a `cotizaciones/<slug>/Cotizacion.xlsx` y trabaja sobre la copia. **Nunca modifica la plantilla maestra.**

- **Nomenclatura corta** (vigente desde 2026-08-06): Excel tiene un límite de longitud de ruta más estricto que el de Windows (260 caracteres); una ruta de unos 236 caracteres falla con «no hemos encontrado el archivo» aunque el archivo sea válido. Por eso los nombres de archivo y carpeta son cortos y sin tildes, y la ruta completa se mantiene **bajo unos 180 caracteres**. Para un cliente con varios proyectos se usa `cotizaciones/<slug-cliente>/<slug-proyecto>/`, con archivos simplemente `Cotizacion.xlsx` y `Cotizacion de Alcance.pdf` (el cliente y el proyecto ya están en la ruta).

El flujo de trabajo



1. **Recibir.** El usuario comparte los alcances (texto, notas, brief) y el nombre o razón social del cliente.
2. **Preguntar ante la duda.** Si hay ambigüedad en el alcance, en el esfuerzo o en cualquier dato necesario para cotizar, el agente **pregunta directamente** antes de continuar.
3. **Generar el PDF.** Sigue la plantilla visual, organizado por módulos. En esta etapa el valor de cada fila aún dice «Pendiente de costear».
4. **Llenar el XLSX.** Con el PDF validado por el usuario, usa el desglose de módulos, submódulos y detalles para llenar la plantilla.
5. **Recalcular y poner precios.** Recalcula el XLSX y **actualiza el PDF** reemplazando «Pendiente de costear» por los precios reales.

6. **Verificar.** Comprueba que ambos archivos reflejen el mismo alcance, que el XLSX recalcule sin errores, que ningún día sea menor a 0,5 ni esté en la celda equivocada, y que los precios del PDF coincidan con el XLSX.
7. **Entregar** ambos archivos y actualizar `wiki/index.md` y `wiki/log.md`.

La plantilla XLSX

La plantilla maestra (hoja `Plantilla`) ya trae **todas las fórmulas de costeo montadas** por el equipo Campus: las columnas A y P a AC, la fila 121 en adelante, etc. El agente solo llena dos tipos de celda.

La jerarquía L / M / N

Desde el 2026-08-05 la jerarquía es **flexible**: la cantidad de funcionalidades (N) bajo cada submódulo (M) es libre, tantas como el alcance real necesite. Queda prohibido forzar «un submódulo, una descripción».

Columna	Nivel	Contenido	Cardinalidad
L	Módulo	`NOMBRE DEL MÓDULO - descripción breve`, combinados en una sola celda, separados por `-`	Una fila por módulo
M	Submódulo	Encabezado corto de una agrupación temática; **sin días propios**	Varios por módulo
N	Funcionalidad	Frase autocontenida que describe **una** capacidad atómica, con sus días en B:K	Varias por submódulo (mínimo 1)

Cada fila L, M o N ocupa su propia fila; las otras dos columnas de esa fila quedan vacías. Un ejemplo de estructura correcta:

```
L: MÓDULO A - descripción breve
M: Submódulo 1
N: Funcionalidad 1 (con sus días)
N: Funcionalidad 2 (con sus días)
N: Funcionalidad 3 (con sus días)
M: Submódulo 2
N: Funcionalidad 1 (con sus días)
L: MÓDULO B - descripción breve
```

```
M: Submódulo 1
N: Funcionalidad 1 (con sus días)
```

El criterio de cardinalidad es una pregunta: ¿cuántas funcionalidades atómicas y realmente distintas describe el alcance para ese submódulo? Ni una menos (no resumir dos funcionalidades reales en una fila), ni una de más forzada para aparentar detalle. En la columna L **no hay numeración** `M1.` o `M2.` (a diferencia del PDF).

Filas fijas y zonas de la hoja

Zona	Regla
Filas 2 a 5	Cuatro categorías transversales **siempre se llenan**: «Estructura del proyecto», «UX y esquema del proyecto», «Implementación y entrega del sistema» y «Estructura del agente IA». No se renombran ni se borran; sus días se estiman por el tamaño y la complejidad general del proyecto
Filas 6 y 7	Quedan siempre vacías (espacio reservado de la plantilla)
Desde la fila 8	El desglose del cliente
Columnas A, O:AC y filas 121 a 140	**Intocables**: fórmulas, checkboxes «Plus/Quitar/2da Ver», resumen y tabla de costos por especialidad
`O1` y `W1`	No se rellenan salvo pedido explícito (descripción del proyecto y fecha)

Las dos celdas autorizadas

Son las **únicas** excepciones a la zona intocable. Se ubican en relación con la «barra gris», la fila con relleno gris sólido (`#999999`) en la columna L que cierra la tabla de alcance. En la plantilla maestra está en la fila 120, pero **se desplaza** si se insertaron filas por un alcance grande, y ambas celdas se mueven con ella.

Celda	En la plantilla maestra	Contenido obligatorio
Columna Y, fila `barra gris + 2`	`Y122`	Fórmula `=Y1/0.6`

Celda	En la plantilla maestra	Contenido obligatorio
Columna R, fila `barra gris + 3`	`R123`	Valor `0.1` (AIU de 10 %), que reemplaza el 40 % por defecto de la plantilla

Antes de escribir en cualquiera de las dos, el agente confirma que esté vacía o con el valor esperado, y confirma `R` por la etiqueta `AIU` de la columna Q de esa fila, no solo por el número. Si ya contiene datos de una fila real de alcance, **no la sobrescribe**: reporta el conflicto.

Si el alcance necesita más filas

Se **insertan filas** (nunca se sobrescribe la zona de fórmulas de resumen) y se verifica que las fórmulas de las columnas A, X, Y, AB y AC se hayan propagado en las filas nuevas antes de recalcular. Esto ya ocurrió en cotizaciones grandes, donde también hubo que reparar referencias de validaciones de datos y formatos condicionales que apuntaban a las filas límite.

Aclaraciones sin celda propia

Si el usuario pide dejar una condición sobre una fila o módulo («este cobro aplica solo si…») y no hay dónde escribirla sin tocar valores o fórmulas, se usa un **comentario de celda de Excel** sobre la celda L o M relevante. No cambia ningún valor ni fórmula, es visible al pasar el cursor y sobrevive al recálculo con LibreOffice.

La estimación de días

Las columnas B a K corresponden a estas especialidades:

Columna	Código	Especialidad
B	DB	DB, Líder, Arquitecto, Scrum
C	UX	Diseño UI/UX
D	FM	FrontEnd Medium
E	BS	BackEnd Senior
F	BM	BackEnd Semi-Senior

Columna	Código	Especialidad
G	QA	QA, Test
H	IM	Implementación, Capacitación
I	MS	Mobile Semi-Senior
J	MM	Mobile Medium
K	IA	BackEnd IA

Las reglas de estimación, vigentes desde el 2026-08-05:

- **El estimado va siempre en la fila de la funcionalidad (N).** El submódulo (M) nunca lleva días: es un encabezado de agrupación, no una unidad de esfuerzo.
- **Cada N tiene su estimado independiente.** Nunca se reparte ni se promedia un solo estimado entre varias filas.
- **Solo se llenan las especialidades que participan.** Las demás quedan vacías, **no en 0**.
- **Piso mínimo de 0,5.** Si la estimación real da menos (por ejemplo 0,23), se redondea a 0,5; por encima, se puede usar cualquier decimal realista (0,8; 1,5; 2,5).
- **Se estima como lo haría un desarrollador real:** complejidad técnica, integraciones externas y alcance descrito, nunca un valor genérico por defecto.

Cuando la fuente ya trae esfuerzo en horas por rol, la conversión usada en entregas anteriores fue **horas ÷ 8 = días**, con el piso de 0,5 por especialidad, repartiendo las horas entre funcionalidades atómicas.

El PDF de cotización

Diseño

El diseño vigente desde el 2026-07-17 usa la **paleta clara de marca** con el contenido en un **grid con líneas delgadas**, tanto horizontales como verticales, como una hoja de cálculo.

Elemento	Especificación
Colores	Azul navy `#152F5E`, azul cielo `#418BF3` y blanco

Elemento	Especificación
Logo	Horizontal azul, arriba a la izquierda, sobre fondo claro
Tabla	Tres columnas: Módulo / Submódulo, Detalle técnico y Valor (COP), en ese orden fijo
Encabezado y total	Fondo navy con texto blanco
Banda de módulo	Fondo azul cielo con texto blanco (`M1. NOMBRE DEL MÓDULO` y su descripción)
Filas	Submódulo en negrita, sin precio propio; funcionalidades con viñeta `•` y precio propio
Tipografía	Arial: módulo 11 pt negrita, submódulo 10 pt negrita, funcionalidad y detalle 10 pt regular

> ESTE ES EL DISEÑO DE LAS COTIZACIONES, NO EL DE LAS PRESENTACIONES
>
> La identidad de marca general de Campuslands tiene su propia paleta y sistema, que se usa en otros agentes. El PDF de cotización mantiene su diseño propio, con grid de líneas finas. En casos puntuales, un cliente pidió una variante (por ejemplo, fondo blanco con barras en los azules corporativos); queda como excepción de ese cliente.

Contenido

- **100 % verídico:** solo lo que el usuario compartió o confirmó. Cero relleno genérico.
- **Espejo del XLSX:** el PDF nunca muestra menos ni más nivel de detalle que el cargado en el Excel. Módulo es L, submódulo es M y funcionalidad es N.
- **Atomicidad de bullets compuestos** (desde 2026-08-06): un bullet de la fuente que une varias cosas con «y» o comas se divide **solo si cada una es una capacidad distinta** (verbos o acciones diferentes). No se divide si enumera dimensiones de una sola capacidad: «por series, subseries, expedientes y dependencias» es una taxonomía, una funcionalidad.
- **Nota de contexto técnico:** si la fuente trae «Componente tecnológico principal» o «Clasificación» por producto, se agrega como nota informativa en cursiva gris bajo el objetivo, sin precio y fuera de la tabla. Se omite si la fuente no los trae.

Precios en el PDF

Después de llenar el XLSX, el agente reemplaza «Pendiente de costear» por los precios reales:

1. Recalcula el XLSX con `recalc.py`, con cero errores fuera de `A140:A144`.
2. **Lee** los precios por fila desde la columna A del XLSX ya recalculado (`data_only=True`): nunca los calcula ni los transcribe a mano.
3. Hace una **doble verificación** antes de entregar:
   - Cada precio de línea del PDF coincide exactamente con su celda de origen.
   - La suma de los ítems de cada módulo coincide con el subtotal de su banda, y la suma de todos los módulos (incluidas las filas transversales 2 a 5) coincide exactamente con el TOTAL del XLSX.

> DIFERENCIAS DE UN PESO
>
> Por redondeo, la suma de las líneas puede diferir en $1 del subtotal de un módulo o del total. El PDF muestra el total exacto del XLSX y lo anota; **nunca se ajusta a mano**.

Verificación obligatoria

Comprobación	Qué se confirma
Recálculo	Cero errores de fórmula, salvo el defecto preexistente de `A140:A144`
Cobertura	Ningún ítem del PDF sin su fila en el XLSX, ni al revés, a nivel de módulo, submódulo y funcionalidad
Días	Cada funcionalidad (N) tiene los suyos; ningún submódulo (M) los tiene; ninguno menor a 0,5
Celda `Y`	`Y{barra gris+2}` tiene `=Y1/0.6` y recalcula a un valor numérico
Celda `R`	`R{barra gris+3}` tiene `0.1` y se refleja en la columna P (valor hora) de la tabla de costos
Precios	Cada línea del PDF coincide con su celda; subtotales y total cuadran con el XLSX

El defecto preexistente de `A140:A144` es un `#NAME?` o `#ERR520` causado por una fórmula `UNIQUE()` que LibreOffice no soporta. Está también en la plantilla maestra en blanco y no lo introduce el llenado del agente.

Entorno y herramientas

La carpeta no incluye scripts propios: el agente se apoya en herramientas del entorno.

Herramienta	Uso
`openpyxl`	Leer y escribir el XLSX, incluidos los comentarios de celda
LibreOffice y `recalc.py` (de la skill de xlsx)	Recalcular las fórmulas y detectar errores
`reportlab`	Generar el PDF a partir de la estructura y los precios leídos del XLSX recalculado

Dos trampas en Windows

Registradas en la bitácora y que conviene conocer:

1. **PATH.** Agregar LibreOffice al **inicio** del `PATH` hace que su Python sombree al del sistema (que tiene `openpyxl`). Hay que agregarlo al **final**.
2. **`soffice.py`.** El script de la skill falla en Windows por `socket.AF_UNIX`; se resolvió con un envoltorio que reemplaza `get_soffice_env` y `run_soffice` antes de ejecutar `recalc.py`, sin tocar la skill.

Además, LibreOffice reescribe al recalcular los booleanos de la columna AB (los checkboxes) como `=TRUE()` y `=FALSE()`. Tienen el mismo valor, por lo que no se corrige.

Comandos «!»

Los comandos son órdenes cortas del usuario (`!optimize`, `!notime`…) que el agente debe ejecutar de forma obligatoria, siempre dentro de sus reglas duras. Hay un solo catálogo para todos los agentes y de él salen las instrucciones de cada uno, las bitácoras y las tablas de las guías.

Dónde vive cada cosa

Qué	Dónde
Catálogo, regla común y qué comandos tiene cada agente	`resources/comandos/comandos.py` (la fuente única; no se edita a mano nada de lo que genera)
Catálogo legible, matriz por agente y decisiones	`resources/comandos/catalogo.md`
Instrucciones del agente	sección «Comandos (`!`)» de `CLAUDE.md`, entre las marcas `<!-- comandos:inicio -->` y `<!-- comandos:fin -->`
Tabla de esta guía	se genera del mismo catálogo al compilar las guías

La regla común

1. Un comando es una palabra precedida de `!` al inicio de un mensaje o de una línea; el texto que sigue es su objeto.
2. Es obligatorio: el agente no pide permiso ni lo reinterpreta. Solo lo limita una **regla dura** (veracidad, marca, confidencialidad, puertas de aprobación y de verificación); entonces hace lo que la regla permite y dice qué no pudo.
3. Se combinan. Si dos modos chocan, el orden es: reglas duras > `!strict-mode` y `!no-assumptions` > `!think-deep`, `!critical-mode` y `!double-check` > `!notime`.
4. Rige la tarea en curso, no las siguientes.
5. La primera línea de la respuesta nombra los comandos recibidos.
6. Un comando desconocido no se adivina: se propone el más parecido y se pregunta. `!help` lista los disponibles.
7. Si un comando cambió un entregable o la wiki, se anota en `wiki/log.md`.

Cómo añadir o cambiar un comando

1. Edita `CMDS` (definición), `AGENTES` (a quién aplica y su nota) o `ALIAS` en `resources/comandos/comandos.py`.
2. Corre `python -X utf8 resources/comandos/comandos.py`: reescribe la sección de cada agente, el catálogo y, la primera vez, añade la entrada a su bitácora.
3. Recompila las guías (`python guides/_src/build.py`).
4. `python resources/comandos/comandos.py --verificar` falla si algo quedó desfasado.

Comandos de este agente

General

Comando	Qué hace aquí
`!analyze`	Analiza la solicitud, el problema, los documentos o la situación **antes de actuar**: qué se pide, qué hay, qué falta y qué riesgos ve. Entrega hallazgos; no ejecuta ni modifica nada. **Aquí:** Analiza los alcances: módulos, funcionalidades y lo que falta para estimar; no llena nada todavía.
`!research`	Investiga a fondo el tema, cruza fuentes y entrega lo relevante con **cada dato citado a su fuente**. Sin fuente verificable no es un dato: se marca `[PENDIENTE]`.
`!plan`	Entrega un plan de trabajo estructurado (pasos, orden, insumos, entregables, supuestos y puntos de decisión) y **se detiene a esperar aprobación** antes de ejecutarlo. **Aquí:** Presenta la estructura de módulos/funcionalidades y el criterio de estimación; espera aprobación antes de llenar el XLSX.
`!execute`	Ejecuta ya lo solicitado o el siguiente paso aprobado del flujo, sin volver a proponer ni a preguntar lo ya decidido. No se salta las puertas de aprobación ni de verificación.
`!review`	Revisa el trabajo indicado y entrega oportunidades de mejora priorizadas (qué cambiaría y por qué). No lo modifica salvo que se le pida.
`!audit`	Auditoría integral de punta a punta: contradicciones entre documentos, cifras que no cuadran, reglas incumplidas y datos sin fuente. Entrega hallazgos con evidencia y severidad.
`!validate`	Comprueba, requisito por requisito, que el resultado cumple lo establecido (reglas duras, insumos, formato). Veredicto «cumple / no cumple» por ítem, con evidencia. **Aquí:** Verificación obligatoria: suma de módulos + transversales = total del XLSX = total del PDF, fórmulas intactas y sin errores nuevos.
`!optimize`	Optimiza lo que se esté desarrollando o ya existe (proceso, solución o resultado): mejora claridad, calidad, consistencia, peso o rendimiento **sin cambiar su alcance ni sus datos**. **Aquí:** Ordena y afina la estructura del XLSX y el PDF sin cambiar días ni precios.
`!compare`	Compara alternativas con criterios explícitos y recomienda la más conveniente, con el porqué.

Comando	Qué hace aquí
`!summarize`	Resumen claro y estructurado (lo esencial, decisiones y pendientes) del trabajo, de un documento o de la conversación.
`!document`	Documenta lo hecho: actualiza la wiki y la bitácora del agente, o genera la documentación técnica, funcional o ejecutiva que se pida.
`!prioritize`	Clasifica actividades, hallazgos o elementos por impacto, importancia y urgencia, y entrega el orden recomendado.
`!help`	Lista los comandos disponibles en este agente, agrupados, con su función.

Modos

Comando	Qué hace aquí
`!notime`	**Modo prisa:** apresura el desarrollo optimizándolo: agrupa preguntas en un solo mensaje, paraleliza, reutiliza lo ya validado, omite explicaciones y pasos opcionales y responde solo con lo esencial. **Nunca** se salta verificaciones obligatorias, puertas de aprobación ni la veracidad: acelera el camino, no recorta la calidad.
`!think-deep`	Análisis más profundo antes de responder: considera alternativas, casos límite y consecuencias, y muestra solo el razonamiento que importa.
`!critical-mode`	Modo crítico: cuestiona decisiones, supuestos y su propio trabajo; busca errores y contradicciones antes de entregar y los dice.
`!no-assumptions`	No supone: ante un dato faltante o ambiguo pregunta (todas las dudas juntas) en lugar de rellenar o inferir.
`!double-check`	Segunda verificación independiente de su propio resultado (releer la fuente, recalcular, comparar) y corrección de lo que encuentre; reporta qué verificó. **Aquí:** Recalcula los totales del XLSX y los compara con el PDF, cifra por cifra.
`!strict-mode`	Sigue al pie de la letra los requisitos e instrucciones dados: sin iniciativas, adiciones ni interpretaciones creativas.

Técnico y seguridad

Comando	Qué hace aquí
`!debug`	Identifica la causa de un error de software (script, conversión, verificación) y lo corrige; vuelve a correr la verificación para comprobarlo.
`!troubleshoot`	Investiga un problema técnico y determina su **causa raíz**, con evidencia. Diagnostica; no cambia nada.
`!security-check`	Revisa seguridad y confidencialidad: datos de otros clientes mezclados, credenciales, datos personales, información que no debe salir del proyecto.

UX/UI y diseño

Comando	Qué hace aquí
`!visual-audit`	Examina proporciones, distribución, tipografía y consistencia visual del entregable, a la vista y página por página. **Aquí:** Revisa a la vista el PDF de cotización (diseño vigente, tablas, paginación).
`!consistency-check`	Detecta inconsistencias entre secciones, componentes, cifras, nombres y estilos del mismo entregable.

Gestión de proyecto

Comando	Qué hace aquí
`!breakdown`	Divide un proyecto o solicitud compleja en actividades específicas, cada una con su entregable y su orden.
`!estimate`	Estima tiempos, recursos y complejidad, con el método y los supuestos escritos (un número sin método no se entrega). **Aquí:** Días por especialidad y funcionalidad, con el método y los supuestos escritos; respeta las celdas y fórmulas intocables de la plantilla.
`!track-progress`	Informa el progreso y los pendientes (hecho, en curso, bloqueado, `[PENDIENTE]`) a partir de la bitácora y del trabajo actual.
`!risk-analysis`	Identifica riesgos, su probabilidad e impacto, y propone la mitigación de cada uno. **Aquí:** Riesgos de la estimación: funcionalidades ambiguas, dependencias, supuestos de precio.

Comando	Qué hace aquí
`!requirements`	Extrae y estructura requerimientos funcionales y no funcionales de las fuentes, cada uno con su referencia de origen. **Aquí:** Extrae de los alcances la jerarquía módulo → submódulo → funcionalidad con su referencia.
`!generate-report`	Genera un informe ejecutivo (decisiones, resultado, pendientes) o técnico (método, evidencias, verificaciones) del trabajo.
`!final-delivery`	Prepara y valida la entrega final: corre todas las verificaciones obligatorias, completa el checklist de cierre (wiki y bitácora) y entrega solo si todo pasa. **Aquí:** XLSX y PDF verificados, precios coherentes entre ambos, wiki y bitácora al día.
`!agent-handoff`	Prepara el traspaso al siguiente agente del flujo comercial: resumen, entregables con su ruta, decisiones tomadas y pendientes. **Aquí:** Prepara el traspaso a **Perfeccionador de Contratos** (`contract-perfectionist`): cotización aprobada y su ruta.

IA avanzada

Comando	Qué hace aquí
`!second-opinion`	Un subagente independiente evalúa el resultado **sin ver tu razonamiento**; se contrasta con el tuyo y se explican las diferencias.
`!cross-review`	Otro especialista (subagente con otro rol) revisa el resultado y devuelve observaciones; se aplican las válidas y se informa de las descartadas.

Multiagente

Comando	Qué hace aquí
`!orchestrate`	Reparte partes **independientes** del trabajo entre varios subagentes en paralelo y las integra en un solo resultado verificado.
`!team-review`	Varios revisores con roles distintos evalúan por separado el mismo resultado y se consolida su evaluación. **Aquí:** Revisores con rol **costos (XLSX)**, **alcance (PDF frente a la fuente)** y **marca y diseño**.
`!consensus`	Consolida las evaluaciones disponibles (de subagentes o de revisiones previas) en una sola recomendación, con las discrepancias a la vista.

Mantenimiento

La bitácora `wiki/log.md` es append-only y cada entrada empieza con `## [YYYY-MM-DD] <tipo> | <descripción>`, con tipos `setup`, `build`, `ajuste`, `deploy` y `lint`. En el lint periódico se revisan contradicciones, información desactualizada, páginas huérfanas y convenciones de diseño que ya no se usan.

Quiero...	Qué tocar
Cambiar una regla del XLSX	`CLAUDE.md` § Reglas del XLSX y su espejo `wiki/plantilla-xlsx.md`
Cambiar el diseño del PDF	`CLAUDE.md` § Reglas del PDF, `wiki/marca-campuslands.md` §5 y `wiki/diseno-pdf-cotizacion.md`
Cambiar la jerarquía o la tipografía	Los tres anteriores y `wiki/perfil-usuario.md`
Registrar una cotización nueva	La carpeta del cliente, `wiki/index.md` y `wiki/log.md`
Actualizar la plantilla maestra	`recursos/FullServices - Plantilla Cotizaciones.xlsx` por el usuario, y revisar el mapeo de celdas

Problemas frecuentes

Síntoma	Causa probable	Qué hacer
`#NAME?` en `A140:A144`	`UNIQUE()` no soportado por LibreOffice	Es preexistente de la plantilla; no se corrige
Excel «no encuentra el archivo»	Ruta demasiado larga	Usar la nomenclatura corta y mantener la ruta bajo unos 180 caracteres
Subtotal con $1 de diferencia	Redondeo	El PDF usa el total exacto del XLSX y lo anota

Síntoma	Causa probable	Qué hacer
Un submódulo aparece con días	Se cargaron en M en lugar de N	Mover los días a las filas N correspondientes
Fórmulas rotas tras insertar filas	No se propagaron A/X/Y/AB/AC	Reparar las fórmulas de las filas nuevas y volver a recalcular
`recalc.py` falla en Windows	PATH de LibreOffice o `socket.AF_UNIX`	Aplicar las dos soluciones descritas arriba
El total del deck de presentaciones no coincide	Se tomó el total del PDF de alcance	Usar la celda `Y122` del XLSX, que es la cifra comercial

Cosas que no están alineadas

Varias referencias de `CLAUDE.md` y de la wiki apuntan a archivos que hoy no existen en esta carpeta:

- Cita `scripts/recalc.py`, pero la carpeta no tiene `scripts/`: `recalc.py` viene de la skill de xlsx.
- Cita `wiki/flujo-trabajo.md`, que no existe; el flujo vive en `CLAUDE.md`.
- `wiki/plantilla-xlsx.md` toma `cotizaciones/comultrasan/Financiera Comultrasan - Cotizacion.xlsx` como referencia de cardinalidad variable; ese archivo sí existe en esta carpeta.

Referencia rápida de archivos

Archivo	Para qué lo abres
`CLAUDE.md`	Reglas del PDF y del XLSX, flujo y nomenclatura
`wiki/plantilla-xlsx.md`	Mapeo de celdas de la plantilla
`wiki/diseno-pdf-cotizacion.md`	Diseño del PDF: paleta, grid y tipografía
`wiki/perfil-usuario.md`	Reglas de oro: veracidad y granularidad fiel

Archivo	Para qué lo abres
`wiki/index.md`	Catálogo y resumen de cotizaciones entregadas
`wiki/log.md`	Historial de builds, ajustes y lint
`recursos/FullServices - Plantilla Cotizaciones.xlsx`	Plantilla maestra, solo lectura