**0) Prompt base (antialucinación + estilo)**

**Úsalo al inicio de cada fase; ya lo he aplicado en lo que sigue.**

**ROL:** Actúa como *arquitecto de software / PM técnico / QA lead / DevOps engineer* según la fase.  
**CONTEXTO:** Proyecto = **InfoMensajero**. Dominio = *mensajería unificada + productividad personal*. Equipo = *1–3 devs Python + 1 QA*. Horizonte = *90 días (MVP + Fase 2)*.  
**REGLAS:**

* No inventes datos faltantes; primero lista **Preguntas pendientes**.
* Si hay ambigüedad, ofrece 2–3 opciones y recomienda 1 con razones.
* Entrega en Markdown con secciones y tablas. Incluye **checklist** final.  
  **FORMATO DE SALIDA:** bullets + tablas + Mermaid cuando aplique.

**1) Visión y alcance (qué y para quién)**

**Statement de problema (1 párrafo)**

Profesionales y equipos pequeños pierden foco alternando entre múltiples apps (WhatsApp, Telegram, Slack, Gmail/Outlook), sin vistas unificadas para notas, tareas y calendario. InfoMensajero centraliza esos servicios en una app de escritorio ligera, con sesiones persistentes, **Kanban** y **Gantt tabular por día**, búsqueda local (FTS) y bóveda cifrada, manteniendo privacidad (local-first).

**Objetivos SMART**

1. **Centralizar** 6+ servicios web con sesiones persistentes en ≤ **2 semanas** de desarrollo efectivo.
2. **Productividad**: crear **Kanban** + **Gantt tabular por día** con CRUD completo y exportación PDF/PNG antes del **día 30**.
3. **Búsqueda local** (FTS5) en notas/kanban y texto pegado con resultados < **150 ms** (p95) en datasets de 5k ítems.
4. **Seguridad**: guardar 100% de secretos en bóveda **scrypt + AES-GCM**; 0 secretos en texto plano.
5. **Usabilidad**: NPS ≥ **+40** en 10 testers internos al finalizar el MVP (día 45).

**Usuarios y Jobs-to-be-done**

| **Usuario** | **Job principal** | **Dolor actual** | **Resultado esperado** |
| --- | --- | --- | --- |
| Profesional individual | Unificar mensajería y tareas | Cambios constantes de app | Una sola ventana con tabs + Kanban |
| PM/Analista | Planificar y comunicar | Falta de línea de tiempo | Gantt diario exportable |
| Soporte/Comercial | Responder rápido | Búsqueda lenta en apps | FTS local + respuestas asistidas (IA opcional) |

**Alcance v1 (In/Out)**

* **IN:** Servicios web embebidos (perfiles aislados), Notas, **Kanban + Gantt diario**, Búsqueda FTS (notas/kanban/pegado), Bóveda, Tema oscuro, Métricas básicas.
* **OUT (v1):** Conectores OAuth (Gmail/Slack), sincronización nube, plugins, multiusuario con roles.

**10 riesgos + mitigación**

| **Riesgo** | **Prob x Impacto** | **Mitigación** | **Plan B** |
| --- | --- | --- | --- |
| Cambios CSP en servicios | M x M | abrir en WebView estándar, sin inyección | “Abrir en navegador externo” |
| RAM por múltiple WebView | M x M | lazy-load y cierre de vistas inactivas | Límite de vistas simultáneas |
| Índice FTS crece | M x M | retención y purga | exportar/archivar |
| Fallos de cifrado por mal uso | B x A | cryptography + pruebas | bloquear bóveda si falla |
| QA insuficiente | M x M | smoke e2e y unit tests | beta cerrada |
| Curva de aprendizaje | B x M | UI minimal + shortcuts | onboarding guiado |
| Export PDF en Windows | M x M | Matplotlib/Qt print framework | Export PNG + impresión del SO |
| Bugs en Gantt tabular | M x M | suite de casos Kanban→Gantt | fallback a lista |
| Latencia FTS en equipos viejos | B x M | índices y límites de página | degradar a búsqueda básica |
| Seguridad de passphrase | M x A | keyring opcional + timeout | exigir passphrase cada arranque |

**Preguntas pendientes:**

* ¿Volumen esperado de tareas activas (pico/mes)?
* ¿Preferencia de exportación: PDF con vector o PNG de alta resolución?

**Checklist (fase 1):** vision OK • SMART OK • scope OK • riesgos OK.

**2) Descubrimiento de usuarios y casos de uso**

**Personas resumidas**

| **Persona** | **Metas** | **Dolores** |
| --- | --- | --- |
| Steve (PM-indie) | Planificar y reportar | Saltos de contexto, falta de timeline |
| Analista de Costos | Documentar, coordinar | Fragmentación de chats/info |
| Soporte Comercial | Responder rápido | No hay búsqueda unificada |

**Top 8 casos de uso**

1. Abrir servicios y conservar sesión.
2. Crear tareas en Kanban y fechas.
3. Ver **Gantt diario** para la semana/mes.
4. Exportar Gantt a PDF/PNG.
5. Tomar notas rápidas y buscarlas.
6. Búsqueda global (FTS) en notas/kanban/pegado.
7. Guardar claves API en bóveda.
8. Métricas de tiempo por servicio.

**Mapa de valor (dolor → función → resultado)**

* Salto de apps → Tabs de servicios → Menos fricción.
* Falta de línea de tiempo → **Gantt tabular diario** → Planeación visual.
* Búsqueda dispersa → FTS local → Respuestas rápidas.

**Criterios de éxito**

* **North Star:** % de jornada con **info centralizada** en InfoMensajero (>60%).
* **Operativas:** (1) Tiempo medio de cambio de servicio < 200 ms, (2) p95 búsqueda < 150 ms en 5k registros, (3) 0 secretos en texto plano.

**3) Requisitos funcionales (MoSCoW)**

| **Prioridad** | **Requisito** | **Reglas de negocio** | **Dependencias** |
| --- | --- | --- | --- |
| **M** | Tabs de servicios con perfiles | un perfil por servicio (ruta única) | UI base, DB services |
| **M** | **Kanban 3 columnas** | mover entre columnas preserva fechas | DB kanban |
| **M** | **Gantt tabular por día** | colorear celdas [inicio, fin]; estados o responsable por color | Kanban con fechas |
| **M** | Búsqueda FTS (notas/kanban/pegado) | stemming/español; 50 resultados | DB+FTS |
| **M** | Bóveda scrypt+AES-GCM | sin passphrase no se lee | DB credentials |
| **S** | Export Gantt PDF/PNG | mantener colores y cabecera | Módulo Gantt |
| **S** | Métricas de uso | foco/blur por servicio | UI base |
| **C** | IA (resumir/traducir/sugerir) | opt-in; texto seleccionado | Bóveda (API key) |
| **W** | Conectores OAuth | ingestión + FTS mensajes | Bóveda + jobs |

**4) Requisitos no funcionales (NFRs)**

| **Categoría** | **Meta medible** |
| --- | --- |
| Rendimiento | cambio de tab < **200 ms**; render Gantt < **400 ms** en 200 tareas x 30 días |
| Seguridad | secretos **solo** en AES-GCM; keyring opcional; bloqueo tras 3 intentos |
| Mantenibilidad | cobertura unit ≥ **65%**; lint sin errores; módulos desacoplados |
| Accesibilidad | contraste AA; teclado completo; tooltips |
| Portabilidad | Windows 10/11; sin drivers extra |
| Observabilidad | logs estructurados; modo debug activable |
| i18n | ES listo; EN opcional |
| Resiliencia | DB WAL; backups locales exportables |

**5) Arquitectura de alto nivel**

**Estilo:** Monolito de escritorio modular (PySide6).  
**Integraciones:** Web embebido (QtWebEngine), exportaciones locales.  
**Decisiones ARQ (ADR) resumidas**

* **ADR-01**: PySide6 + QtWebEngine (pros: nativo; contras: footprint).
* **ADR-02**: SQLite+FTS5 (pros: local, rápido; contras: locking si mal uso).
* **ADR-03**: Gantt **tabular** (QTableWidget) en vez de gráfico (pros: editable, familiar; contras: menos “bonito”).
* **ADR-04**: Bóveda scrypt+AES-GCM (pros: estándar; contras: UX de passphrase).
* **ADR-05**: IA opt-in con proveedor intercambiable.

**Diagrama (contexto)**

flowchart LR

User-->UI[PySide6 UI]

UI-->WebViews[QtWebEngine (perfiles)]

UI-->DB[(SQLite + FTS5)]

UI-->Vault[scrypt + AES-GCM]

UI-->Gantt[QTable Gantt Diario]

**6) Modelo de datos (clave para Kanban→Gantt)**

**ERD (resumen)**

erDiagram

SERVICES ||--o{ TABS : has

KANBAN\_COLUMNS ||--o{ KANBAN\_CARDS : has

KANBAN\_CARDS {

int id PK

int column\_id FK

text title

text description

text assignee

datetime created\_at

datetime start\_date

datetime due\_date

datetime completed\_at

text state -- todo|doing|done (derivado de columna)

}

MESSAGES\_INDEX {

text source

text thread\_id

text author

text snippet

text content

datetime created\_at

}

CREDENTIALS {

text id PK

blob enc\_blob

blob nonce

blob salt

}

**Diccionario (campos nuevos para Gantt):**  
start\_date (inicio real/previsto), due\_date (vencimiento), completed\_at (si done).  
**Reglas de mapeo a Gantt diario:**

* **todo** ⇒ [created\_at, due\_date] (gris)
* **doing** ⇒ [start\_date, due\_date] (azul)
* **done** ⇒ [start\_date, completed\_at] (verde)

**7) Backlog (historias + Gherkin) — extracto clave Gantt**

**US-GANTT-01 – Ver Gantt diario**

* Como usuario quiero ver un Gantt **por día** generado desde el Kanban para planificar.  
  **Criterios**
* Dado tareas con fechas, cuando abro “Ver Gantt”, entonces veo tabla con columnas = días del rango y celdas coloreadas por estado.

**US-GANTT-02 – Exportar Gantt**

* … cuando pulso “Exportar PDF/PNG”, entonces se guarda un archivo con la misma apariencia.

**DoR:** tareas con start\_date y due\_date definidos (o reglas de inferencia).  
**DoD:** render < 400 ms en 200x30; export OK; pruebas de 3 casos extremos.

**8) Estimación y roadmap (90 días)**

| **Semana** | **Hito** |
| --- | --- |
| 1–2 | Infra UI + Servicios con perfiles + Notas |
| 3–4 | **Kanban CRUD** + fechas; **Gantt diario** (render + export) |
| 5–6 | Búsqueda FTS + Bóveda cifrada |
| 7–8 | Métricas + Tema oscuro + pulido UX |
| 9–10 | IA (opt-in) + beta testers + hardening |
| 11–12 | Ajustes finales, docs, empaquetado e instalador |

Supuesto: 1–2 devs; buffer riesgo: 15%.

**9) Diseño técnico por módulo — foco Gantt**

**Módulo Gantt (QTableWidget)**

* **Entradas:** lista de KANBAN\_CARDS + rango [from\_date, to\_date].
* **Salida:** tabla con cabecera por día y celdas coloreadas (estado/responsable).
* **Errores:** fechas inconsistentes → normalizar y señalar.
* **Patrón:** *Presenter* que transforma tareas en “segmentos por día”.

**Secuencia (Kanban→Gantt)**

sequenceDiagram

User->>UI: Abrir "Ver Gantt"

UI->>DB: SELECT cards con fechas

DB-->>UI: tarjetas

UI->>GanttPresenter: build\_matrix(cards, rango)

GanttPresenter-->>UI: matriz [filas x días] + colores

UI->>QTable: pintar celdas + tooltips

**10) Plan de pruebas (QA)**

| **Tipo** | **Qué validamos** |
| --- | --- |
| Unit | mapeo estado→rango; matriz diaria; export |
| Integración | CRUD Kanban → Gantt render |
| E2E | flujo usuario (abrir, rango, exportar) |
| Rendimiento | render p95 < 400 ms (200x30) |
| Seguridad | bóveda encripta/descifra; sin secretos en logs |

**11) DevOps/CI para app de escritorio**

* **CI (GitHub Actions):** lint (ruff/black) → tests (pytest) → build (PyInstaller) → subir artefacto.
* **Canales:** *alpha* (artefacto CI), *beta* (release manual), *stable* (tag + checksum).
* **Observabilidad local:** logs rotativos, flag --debug.

**12) Seguridad y cumplimiento**

* Modelo STRIDE resumido: *Tampering* (DB) mitigado por WAL + checksums; *Information disclosure* mitigado por bóveda; *Repudiation* no aplica multiusuario.
* Políticas: backup local exportable; retención configurable; wipe seguro de perfiles a pedido.

**13) Documentación viva**

* **README** (run/build), **ADR/**, **RUNBOOK** (preguntas típicas), **CONTRIBUTING**, **CHANGELOG**.
* Plantillas: Historia+Gherkin, ADR, Riesgos, NFR.

**14) Sprint 0**

* Repos y estructura (src/app/...).
* Tooling (ruff, black, mypy opcional, pre-commit).
* “Hello vertical”: cargar un servicio, crear 1 tarea Kanban, render Gantt diario con 7 días, export PNG.

**15) Operación ágil**

* Rituales: planning semanal, dailies cortas, review + retro quincenal.
* Tablero: Backlog → Ready → Doing → In Review → Done.
* Políticas de prioridad: MoSCoW + riesgo/valor.

**Plantillas rápidas (listas para usar)**

**Historia + Gherkin**

* Como [rol] quiero [acción] para [beneficio].
* **Dado** [estado], **cuando** [acción], **entonces** [resultado].

**ADR**

* Título, Contexto, Opciones (pros/cons), Decisión, Consecuencias.

**NFR**

* Categoría, Meta, Métrica, Umbral.

**Riesgo**

* Riesgo, Prob x Impacto, Mitigación, Plan B.

**Preguntas pendientes globales**

1. ¿Rango por defecto del Gantt (7/14/30 días)?
2. ¿Colores por estado o por responsable (preferencia inicial)?
3. ¿Exportación prioritaria: **PDF vector** o **PNG 2×**?
4. ¿Límite de tareas simultáneas esperado?