# Blueprint maestro – InfoMensajero (Python)

**Objetivo:** Aplicación de escritorio para Windows (multiplataforma) que unifica mensajería y productividad con sesiones persistentes, seguridad local y características de IA, desarrollada **solo con Python** y stack Qt (PySide6 + QtWebEngine), SQLite y criptografía estándar.

## 1. Alcance y visión de producto

* **Usuarios objetivo:** profesionales individuales y equipos pequeños que necesitan centralizar WhatsApp, Telegram, Slack, Gmail/Outlook (web), Teams y LinkedIn en un solo entorno de escritorio, con notas, kanban y calendario.
* **Principios:** minimalista, seguro, ligero, sin anuncios, 100% open source. Privacidad primero (local-first, nada se sube sin consentimiento).
* **Resultado:** un “hub” con pestañas/servicios, sesiones persistentes, atajos de productividad, búsquedas locales y herramientas IA opcionales.

**No incluye (fase inicial):** sincronización en la nube, plugins de terceros, conexiones a APIs de pago por defecto. Se dejan preparados puntos de extensión.

## 2. Requerimientos (funcionales y no funcionales)

### 2.1 Funcionales

1. **Servicios embebidos** en pestañas con **perfiles aislados** (cookies/cache por servicio).
2. **Gestión de servicios**: catálogo predefinido + alta de servicios personalizados por URL.
3. **Sesión persistente**: reabrir app conserva login (gracias a perfiles QtWebEngine separados por servicio).
4. **Notas rápidas** y **Kanban** integrados localmente.
5. **Calendario** (Google/Outlook) embebido.
6. **Buscador global (FTS5)** sobre notas/kanban y textos que el usuario pegue o importe.
7. **Seguridad local**: bóveda con scrypt + AES‑GCM para API keys/credenciales.
8. **IA (opcional)**: resumir, traducir, proponer respuestas (proveedor intercambiable, opt‑in).
9. **Métricas de uso** por servicio (tiempo en foco) y **notificaciones silenciosas** configurables.
10. **Tema claro/oscuro**, atajos y UI accesible.

### 2.2 No funcionales

* **Rendimiento:** inicio < 3 s en equipos estándar, cambio de pestaña < 200 ms (objetivo), huella RAM moderada.
* **Seguridad:** cifrado fuerte (scrypt, AES‑256‑GCM), mínimo IPC, zero‑trust por defecto.
* **Portabilidad:** Windows 10/11, con proyección a macOS/Linux.
* **Mantenibilidad:** modular, pruebas automáticas básicas, linters y estilo consistente.
* **Confiabilidad:** persistencia WAL en SQLite, manejo de errores UI clara.

**Criterios de aceptación (MVP):** - Añadir y abrir 5+ servicios, conservar sesión al reiniciar. - Crear/editar 10 notas, 10 tarjetas kanban; búsqueda FTS las encuentra. - Guardar y recuperar una API key en la bóveda. - IA demo funcional (stub o proveedor real con clave del usuario).

## 3. Arquitectura

### 3.1 Capas

* **UI/Presentación (Qt Widgets):** ventanas, sidebar, lista de servicios, notas, kanban, barra superior (búsqueda/IA).
* **Navegador embebido:** QtWebEngine (QWebEngineView) por **servicio** con **QWebEngineProfile** dedicado (ruta de perfil única por servicio).
* **Dominio:** orquestación de servicios, indexación, métricas, seguridad.
* **Datos:** SQLite (WAL), tablas normalizadas + FTS5.
* **Cripto:** scrypt + AES‑GCM; keyring Windows (opcional) para cachear passphrase.

### 3.2 Diagrama lógico (texto)

* **MainWindow** ↔ **Sidebar** (eventos abrir servicio / crear nota / alta servicio)
* **MainWindow** ↔ **WebEngineView** (carga/gestión de perfiles/URLs)
* **DB** (servicios, notas, kanban, índice FTS, usage, credentials)
* **Vault** (guardar/leer secretos) ↔ **DB.credentials** ↔ **keyring** (cache opcional)
* **AI Provider** (interfaz) → implementación elegible (Groq/OpenAI/Local)
* **Indexer** (insertar/consultar FTS5)

## 4. Diseño de módulos (Python)

* app/ui/ – Qt Widgets (MainWindow, Sidebar, diálogos). Estilos (tema) por stylesheet.
* app/services/ – catálogo y manager para alta/listado; asigna profile\_path.
* app/db/ – conexión SQLite, schema, CRUD para entidades.
* app/search/ – indexador y consultas FTS5.
* app/security/ – vault (scrypt + AES‑GCM), keyring opcional.
* app/metrics/ – registro de foco/blur y acumulados por servicio/día.
* app/ai/ – interfaz de IA con proveedor intercambiable.
* app/utils/ – paths, helpers.

**Estructura de carpetas** – ver documento de implementación (canvas) para el árbol completo.

## 5. Modelo de datos (SQLite)

Tablas clave: - **services**(id, name, url, icon, profile\_path, is\_active) - **tabs**(id, service\_id, title, url, pinned, created\_at) - **notes**(id, content, created\_at, updated\_at) - **kanban\_columns**(id, name, position) - **kanban\_cards**(id, column\_id, title, description, created\_at) - **messages\_index** FTS5(source, thread\_id, author, snippet, content, created\_at) - **credentials**(id, enc\_blob, nonce, salt) - **usage\_metrics**(id, service\_id, milliseconds, day) - **settings**(key, value)

**Índices recomendados:** - kanban\_cards(column\_id, created\_at) - usage\_metrics(service\_id, day) - FTS5 ya indexa texto; considerar trigramas si fuera necesario más adelante.

## 6. Seguridad y privacidad

* **Bóveda:** contraseña del usuario → scrypt (N=2^15, r=8, p=1) → AES‑256‑GCM con nonce de 96 bits.
* **Almacenamiento de sesiones:** solo cookies/cache en QWebEngineProfile por servicio. No se copian ni inspeccionan por defecto.
* **Opt‑in IA/APIs:** envío de texto solo bajo acciones explícitas (ej. botón “Resumir selección”). No hay recolección silenciosa.
* **Permisos web:** restringir descargas automáticas, JS injection deshabilitado por defecto.
* **Actualizaciones:** firma del binario (futuro) y checksum del instalador.

Gestión de secretos: - credentials.id por proveedor (p.ej., groq). - Cacheo opcional de passphrase en Windows Credential Manager (keyring).

## 7. UX / UI

### 7.1 Layout principal

* **Header:** buscador global (FTS), botones IA (**/resumir, /traducir, /sugerir**), toggle tema.
* **Sidebar (260 px):** lista de servicios (botón **+ Servicio**), bloque **Notas rápidas** (input + lista), bloque **Kanban** (vista compacta 3 columnas).
* **Área central:** vista web del servicio activo.

### 7.2 Flujos clave

* **Agregar servicio:** + Servicio → Diálogo (Nombre, URL) → crea registro y **profile\_path** → abre pestaña.
* **Persistencia de sesión:** usuario inicia sesión en el servicio web → cookies quedan en profile\_path → al reiniciar, el servicio queda logueado.
* **Búsqueda global:** usuario escribe término → FTS5 entrega resultados (notas/kanban/pegados). Más adelante, incluirá mensajes importados por conectores.
* **IA:** selección de texto → botón IA → popup con resultado (sin subir nada no seleccionado).

Accesibilidad: atajos para cambiar servicio, crear nota, buscar (Ctrl+F global), alto contraste.

## 8. Búsqueda e indexación

* **FTS5** sobre messages\_index, notes, kanban\_cards.
* **Ingesta inicial (MVP):**
  + manual (pegar/exportar textos) → messages\_index.
  + notas y kanban se indexan automáticamente con triggers lógicos (en la capa Python).
* **Futuro:** conectores OAuth (Gmail/Outlook/Slack) para ingestión periódica. Tareas programadas con backoff y límites de tasa.

Ranking básico: FTS5 BM25 default + boosts por campo (snippet > content) si aplica.

## 9. IA integrada (opcional)

Interfaz simple en app/ai/provider.py: - **summarize(text)** – resume con proveedor seleccionado. - **translate(text, lang)** – traducción simple. - **suggest\_reply(context)** – plantillas y/o LLM.

Política: el usuario pega o selecciona texto. Nunca se indexa ni se envía automáticamente sin interacción.

## 10. Métricas y notificaciones

* **Métricas:** on\_focus/on\_blur por servicio → acumular milliseconds por day.
* **Alertas de uso:** reglas sencillas (ej. > 2h continuas) → notificación local discreta.
* **Notificaciones silenciosas:** panel por servicio para habilitar/deshabilitar sonidos/toasts del sitio web.

## 11. Rendimiento

* **Perfiles dedicados** → cachés por servicio (arranques rápidos tras primer login).
* **Lazy load** de servicios (crear perfil al abrirlo por primera vez).
* **WAL** en SQLite para concurrencia y durabilidad.
* **Evitar polling** innecesario; usar timers razonables.

## 12. Empaquetado y despliegue

* **PyInstaller** (Windows) → .exe portable/instalable.
* **Recursos**: incluir catalog.json y assets.
* **Actualizaciones**: manuales en MVP; futuro auto‑update opcional.

## 13. Calidad: testing y tooling

* **Tests unitarios**: db (CRUD), vault (cifrar/descifrar), indexer (FTS), métricas.
* **Tests de humo UI**: lanzar app, abrir servicio, crear nota.
* **Linters**: ruff/flake8, black para formato, mypy opcional.
* **CI**: workflow para lint + tests en PRs.

## 14. Roadmap por fases (sin fechas)

* **Fase 1 (MVP):** servicios embebidos, notas, kanban básico, búsqueda FTS (notas/kanban/pegado), vault, tema oscuro, métricas.
* **Fase 2:** IA con proveedor (opt‑in), atajos completos, notificaciones silenciosas avanzadas, export/backup.
* **Fase 3:** conectores OAuth (Gmail/Outlook/Slack) para indexar mensajes reales; buscador unificado.
* **Fase 4:** multi‑ventana, vistas divididas, reglas de concentración (Do Not Disturb), export analítica.

## 15. Riesgos y mitigaciones

* **Bloqueos de terceros** (CSP, login web): usar user‑agent por defecto, evitar inyección; si un servicio rompe, ofrecer abrir en navegador externo.
* **Consumo RAM** (Chromium): carga diferida y cierre de vistas inactivas.
* **Cripto mal gestionada:** librería probada (cryptography), parámetros scrypt conservadores, pruebas de integración.
* **Índice FTS grande:** purga/rotación y límites de retención.

## 16. Extensibilidad

* **Servicios personalizados:** alta por URL (ya soportado).
* **Conectores**: módulo por proveedor con contrato (pull → normalizar → indexar).
* **Temado**: hojas de estilo con variables (tema claro/oscuro).
* **IA**: registrar nuevos proveedores manteniendo interfaz.

## 17. Entregables

* **Repositorio** con estructura modular Python.
* **Manual de instalación** (requirements, run, build).
* **Guía de contribución** (estándares código, PR, issues).
* **Licencia MIT**.

## 18. Aceptación final (demo)

1. Alta de 3 servicios (WhatsApp, Gmail, Slack) y persistencia de sesión tras reinicio.
2. Creación de 5 notas y 5 tarjetas; búsqueda las encuentra por término.
3. Guardar y recuperar una API key ficticia mediante bóveda.
4. IA demo ejecuta un resumen de un texto pegado.

Este blueprint define la **arquitectura, modelo de datos, seguridad, UX, rendimiento, empaquetado y roadmap** para desarrollar InfoMensajero íntegramente con **Python + PySide6 + QtWebEngine + SQLite**. Sirve como base para planificación, implementación y contribuciones open source.

# 19. Historias de usuario (Product Backlog) + Criterios de aceptación

Formato: **Como [rol] quiero [necesidad] para [beneficio]**. Criterios en estilo Gherkin.

## 19.1 Epic A – Servicios y sesiones persistentes (MVP)

**US-A1 – Agregar servicio desde catálogo**  
Como usuario quiero agregar un servicio del catálogo para abrirlo en la app.  
**Criterios**  
- **Given** la app abierta **When** selecciono “WhatsApp” en el catálogo **Then** se crea el servicio en la barra lateral y se abre la vista. - **Given** un servicio ya agregado **When** reinicio la app **Then** el servicio permanece en la lista.

**US-A2 – Agregar servicio personalizado**  
Como usuario quiero registrar un servicio por URL para integrarlo aunque no esté en catálogo.  
**Criterios**  
- **Given** ingreso Nombre y URL válidos **When** confirmo **Then** aparece en la lista y abre la vista. - **Given** URL inválida **When** intento guardar **Then** veo un error claro y no se crea el servicio.

**US-A3 – Sesión persistente por servicio**  
Como usuario quiero mantener mi sesión iniciada para no reloguear cada vez.  
**Criterios**  
- **Given** inicio sesión en un servicio **When** cierro y reabro la app **Then** sigo logueado. - **Given** borro el perfil del servicio **When** reabro **Then** debo iniciar sesión nuevamente.

**US-A4 – Alternar entre servicios rápidamente**  
Como usuario quiero cambiar de servicio con un clic o atajo.  
**Criterios**  
- **Given** 3+ servicios **When** hago clic en uno **Then** la vista cambia < 200 ms (objetivo) y conserva el estado.

## 19.2 Epic B – Notas rápidas y Kanban (MVP)

**US-B1 – Crear nota rápida**  
Como usuario quiero guardar una nota breve para no olvidar pendientes.  
**Criterios**  
- **Given** escribo en “Notas” **When** presiono Ctrl+Enter **Then** la nota se guarda y aparece al inicio de la lista.

**US-B2 – Ver/Buscar notas**  
Como usuario quiero listar y buscar notas para recuperarlas rápido.  
**Criterios**  
- **Given** notas guardadas **When** abro la app **Then** veo las notas más recientes. - **Given** un término en la barra de búsqueda **When** ejecuto búsqueda **Then** veo coincidencias usando FTS5.

**US-B3 – Kanban 3 columnas**  
Como usuario quiero organizar tareas en Por Hacer/En Curso/Hecho.  
**Criterios**  
- **Given** una nueva tarea **When** la agrego a Por Hacer **Then** queda listada con título. - **Given** una tarea **When** la muevo a Hecho **Then** cambia de columna y persiste.

## 19.3 Epic C – Calendario embebido (MVP)

**US-C1 – Ver Google Calendar**  
Como usuario quiero abrir Calendar dentro de la app.  
**Criterios**  
- **Given** servicio “Google Calendar” **When** lo abro **Then** se muestra calendar.google.com con mi sesión persistente si ya inicié.

**US-C2 – Ver Outlook Calendar**  
Similar a C1 pero con Outlook Web.

## 19.4 Epic D – Búsqueda global (MVP)

**US-D1 – Buscar en notas/kanban y contenido pegado**  
Como usuario quiero buscar en todo mi contenido local.  
**Criterios**  
- **Given** notas y tarjetas **When** escribo “reunión” **Then** veo coincidencias por relevancia (FTS5) con snippet.

## 19.5 Epic E – Seguridad local (MVP)

**US-E1 – Guardar secreto con clave maestra**  
Como usuario quiero guardar una API key cifrada.  
**Criterios**  
- **Given** establezco una passphrase **When** guardo la clave **Then** queda cifrada con AES‑GCM y derivada con scrypt. - **Given** una passphrase incorrecta **When** intento leer **Then** se rechaza y muestra error claro.

**US-E2 – Recordar passphrase (opcional)**  
Como usuario quiero recordar mi passphrase en Windows (Credential Manager).  
**Criterios**  
- **Given** activo “Recordar” **When** reinicio **Then** puedo descifrar sin volver a escribir (hasta que lo desactive).

## 19.6 Epic F – IA (Fase 2)

**US-F1 – Resumir selección**  
Como usuario quiero seleccionar texto en un servicio y obtener resumen.  
**Criterios**  
- **Given** un texto seleccionado **When** pulso “/resumir” **Then** el resumen aparece en un panel lateral. - **Given** IA deshabilitada **When** pulso “/resumir” **Then** la app pide habilitar IA y pegar API key (usa bóveda).

**US-F2 – Traducir selección**  
Criterios similares a F1 con parámetro de idioma.

**US-F3 – Sugerir respuesta**  
Como usuario quiero recibir una respuesta sugerida basada en el contexto que yo aporte.  
**Criterios**  
- **Given** pego un hilo **When** pulso “/sugerir” **Then** recibo un borrador editable.

## 19.7 Epic G – Métricas y notificaciones (Fase 2)

**US-G1 – Métricas de tiempo por servicio**  
Como usuario quiero ver cuánto tiempo paso en cada servicio.  
**Criterios**  
- **Given** uso la app **When** abro Panel de Productividad **Then** veo minutos acumulados por día y servicio.

**US-G2 – Alertas de uso excesivo**  
Como usuario quiero ser alertado al exceder un umbral.  
**Criterios**  
- **Given** umbral 120 min **When** lo supero **Then** recibo una alerta silenciosa.

**US-G3 – Notificaciones silenciosas**  
Como usuario quiero silenciar notificaciones por servicio.  
**Criterios**  
- **Given** desactivo sonido para Slack **When** llegan notifs **Then** no suena pero sigo viendo badges.

## 19.8 Epic H – Conectores APIs (Fase 3)

**US-H1 – Indexar Gmail con OAuth**  
Como usuario quiero autorizar Gmail para buscar mis correos localmente.  
**Criterios**  
- **Given** proceso OAuth exitoso **When** activo indexación **Then** se llenan tablas normalizadas y FTS5.

**US-H2 – Indexar Outlook**  
Análogo a H1 con Microsoft Graph.

**US-H3 – Indexar Slack**  
Análogo a H1 con Slack Web API.

# 20. WBS (Work Breakdown Structure)

Niveles: 1=Épica, 2=Paquete de trabajo, 3=Tarea. Estimaciones orientativas (días hábiles) para un dev full‑stack Python.

## 20.1 Epic A – Servicios & Sesiones (8–12 d)

* **A.1 Infra QtWebEngine** (2 d)
  + A.1.1 Arranque app y ventana principal (0.5 d)
  + A.1.2 Vista web y perfiles por servicio (1 d)
  + A.1.3 Manejo errores carga/red (0.5 d)
* **A.2 Catálogo y alta servicio** (2 d)
  + A.2.1 Cargar catalog.json y bootstrap DB (0.5 d)
  + A.2.2 Diálogo alta personalizada (0.5 d)
  + A.2.3 Persistencia/orden en sidebar (1 d)
* **A.3 Persistencia de sesión** (2 d)
  + A.3.1 Directorios de perfil y limpieza segura (1 d)
  + A.3.2 QA multi‑servicio (1 d)
* **A.4 Navegación/atajos** (2–6 d)
  + A.4.1 Atajos teclado (0.5 d)
  + A.4.2 Estado UI y foco (1 d)
  + A.4.3 Optimización rendimiento (1–4 d)

## 20.2 Epic B – Notas & Kanban (4–6 d)

* **B.1 Notas** (2 d)
  + B.1.1 CRUD notas (1 d)
  + B.1.2 Listado y orden (1 d)
* **B.2 Kanban** (2–4 d)
  + B.2.1 Estructura 3 columnas (1 d)
  + B.2.2 CRUD tarjetas y mover entre columnas (1–3 d)

## 20.3 Epic C – Calendario (1–2 d)

* **C.1 Vistas Calendar/Outlook** (1–2 d)

## 20.4 Epic D – Búsqueda global (3–5 d)

* **D.1 FTS5** (1 d)
* **D.2 Indexación notas/kanban/pegado** (1–2 d)
* **D.3 UI de resultados** (1–2 d)

## 20.5 Epic E – Seguridad (3–5 d)

* **E.1 Vault scrypt + AES‑GCM** (2 d)
* **E.2 Integración keyring opcional** (0.5 d)
* **E.3 Flujos de error y UX** (0.5–2.5 d)

## 20.6 Epic F – IA (4–8 d)

* **F.1 Proveedor IA** (2–3 d)
* **F.2 UI/selección y resultados** (1–3 d)
* **F.3 Configuración y permisos** (1–2 d)

## 20.7 Epic G – Métricas & Notifs (3–6 d)

* **G.1 Hook foco/blur y DB** (1–2 d)
* **G.2 Panel de métricas** (1–2 d)
* **G.3 Silenciamiento por servicio** (1–2 d)

## 20.8 Epic H – Conectores OAuth (10–18 d)

* **H.1 Gmail** (3–6 d)
* **H.2 Outlook/Graph** (3–6 d)
* **H.3 Slack** (4–6 d)

**Estimación total (MVP A–E):** 19–30 días. Con F–G: +7–14 d. Con H: +10–18 d.

# 21. Diagramas de secuencia (texto)

## 21.1 Alta de servicio personalizado

Usuario -> UI: Clic en "+ Servicio"  
UI -> Usuario: Diálogo (Nombre, URL)  
Usuario -> UI: Confirmar  
UI -> ManagerServicios: add\_custom\_service(nombre, url)  
ManagerServicios -> FS: crear profile\_path  
ManagerServicios -> DB: INSERT services(id,name,url,profile\_path)  
DB --> ManagerServicios: OK  
ManagerServicios --> UI: datos servicio  
UI -> Navegador: abrir URL con QWebEngineProfile(profile\_path)  
Navegador --> UI: Vista cargada

## 21.2 Sesión persistente

Usuario -> ServicioWeb: login (dentro del WebView)  
ServicioWeb -> PerfilQt: set cookies/localStorage  
--- reinicio app ---  
UI -> Navegador: abrir servicio con profile\_path existente  
PerfilQt -> ServicioWeb: enviar cookies  
ServicioWeb --> UI: sesión válida, muestra bandeja

## 21.3 Guardado de secreto en bóveda

Usuario -> UI: Pegar API Key + passphrase  
UI -> Vault: save\_secret(id, plaintext, passphrase)  
Vault -> KDF: scrypt(passphrase, salt)  
Vault -> AES-GCM: encrypt(plaintext, key, nonce)  
Vault -> DB: UPSERT credentials(id, enc\_blob, nonce, salt)  
DB --> Vault: OK  
Vault --> UI: OK

## 21.4 Búsqueda global (FTS)

Usuario -> UI: escribir término  
UI -> Search: FTS(query)  
Search -> DB: SELECT ... MATCH query  
DB --> Search: filas  
Search --> UI: render resultados (snippets)

## 21.5 Resumen con IA (opt‑in)

Usuario -> UI: seleccionar texto + "/resumir"  
UI -> ProviderIA: summarize(text)  
ProviderIA -> (Proveedor real o stub): procesar  
(Proveedor) --> ProviderIA: resumen  
ProviderIA --> UI: mostrar resumen

# 22. Plan de pruebas (resumen)

* **Unidad:** db CRUD, vault cifrado/descifrado, FTS index/search, métricas.
* **Integración:** alta servicio → perfil → persistencia sesión; guardar/leer secreto; búsqueda con resultados.
* **UI (smoke):** abrir 3 servicios, crear 3 notas, mover 2 tarjetas, ejecutar una búsqueda y un resumen (stub).

# 23. Checklist de lanzamiento MVP

* Catálogo + alta personalizada operativos
* Perfiles con persistencia confirmada (WhatsApp, Gmail, Slack)
* Notas y Kanban CRUD estables
* Búsqueda FTS en notas/kanban/pegado
* Vault funcional con passphrase y error handling
* Tema oscuro y atajos básicos
* Build PyInstaller sin dependencias faltantes