**PROMPT MAESTRO — CRM de Propuestas/Proyectos con Base de Conocimiento**

**Escritorio (Python + SQLite + Google Drive + Microsoft Graph + Git/GitHub CI/CD + Inno Setup)**

**1) Propósito**

Construir una aplicación de escritorio en **Python** que funcione como **CRM** especializado en propuestas que se convierten en proyectos, con **trazabilidad comercial y técnica completa**, **documentos centralizados en Google Drive**, **colaboración multiusuario** (SQLite central en Drive), **búsqueda global**, **auditoría total**, **gestor de actividades integrado con Outlook**, **Hub de Mensajería** y **checklist por proyecto**.  
**No es** un gestor operativo de proyectos (sin CPM ni dependencias complejas).

**2) Alcance funcional**

**2.1 Autenticación y Autorización**

* Login con usuario/contraseña (hash **Argon2**), sesiones cifradas.
* Roles iniciales: **SuperAdmin**, **Comercial**, **Ingeniero**, **Lectura/Auditor**.
* Permisos base:
  + **Comercial**: crea/le mueve fases de propuestas (únicos autorizados para cambiar de fase), registra interacciones, ve documentos, crea actividades y usa Hub de Mensajería.
  + **Ingeniero**: adjunta/gestiona documentación técnica, línea base, ajustes; **no** mueve fases.
  + **SuperAdmin**: gestiona usuarios/roles, políticas, configuración Drive/BD/Outlook, auditoría, backups.
  + **Lectura/Auditor**: solo lectura y reportes.
* Política de contraseñas (mínimos, bloqueo por intentos, expiración opcional).

**2.2 Leads → Clientes (ciclo de vida de cuentas)**

* **Account** con status ∈ {lead, cliente}.
* Convertir **Lead → Cliente** automáticamente cuando se gana la 1ª propuesta; registrar converted\_at/by.
* En **Clientes**, listar **proyectos históricos** con **fechas de entrega** y **renovación** para campañas de mantenimiento/mejoras.

**2.3 Fases CRM (no PMO)**

* Flujo por defecto **configurable por el usuario**:
  1. Prospección (Comercial)
  2. Calificación (Comercial)
  3. Propuesta (origina documentos de ingeniería; requiere línea base técnica o referencia a doc del cliente/área)
  4. Negociación (ajustes comerciales y de ingeniería)
  5. Ganado (obligatorio registrar: **fecha de cierre**, **fecha de entrega** y **fecha de renovación futura**; confirmar carpeta del proyecto en Drive)
  6. Perdido (motivo de pérdida + lecciones para mejora futura)
* Reglas:
  1. Solo **Comercial** mueve de fase.
  2. Antes de **Ganado/Perdido** debe existir **adjunto técnico** (propuesta/soporte de ingeniería).

**2.4 Propuestas → Proyectos**

* Crear propuesta (cliente/cuenta, título, descripción, condiciones, monto/moneda, validez, estado).
* **Convertir a proyecto** (desde propuesta “aprobada/ganada”):
  + Crear registro **Project** vinculado a la propuesta/cuenta.
  + Crear carpeta en Drive para el proyecto: /ContenedorProyectos/{Año}/{Cliente}-{CódigoProyecto}/.
  + Migrar/relacionar adjuntos y metadatos.

**2.5 Documentos y Google Drive**

* Súper Admin define **enlace/ID de carpeta raíz** y **credenciales** (OAuth de escritorio o Service Account) para:
  + **BD SQLite** centralizada (archivo .db en Drive).
  + **Contenedor de proyectos** (subcarpetas por proyecto).
* Adjuntos (PDF, DOCX, XLSX, imágenes, ZIP…): subir/descargar con progreso; **versionado lógico** (si sube mismo nombre → version += 1); metadatos (nombre, tipo, tamaño, hash, versión, usuario, fecha).

**2.6 Hub de Mensajería (comunicaciones rápidas)**

* **Deep-links** por canal:
  + WhatsApp: https://wa.me/{phone}?text={encoded\_text}
  + Microsoft Teams: msteams:// o URL de reunión
  + Google Meet: https://meet.google.com/new (o existente)
  + Email: mailto:{email}?subject={}&body={}
  + Teléfono/VoIP: tel:{number} o sip:{uri}
* **Plantillas** con variables ({Cliente}, {Proyecto}, {Vencimiento}, {Responsable}, …).
* Registro de interacción (metadatos + enlace externo); evidencias opcionales en Drive.
* **No** se ingiere contenido de chats; se guardan **enlaces/IDs** y **resúmenes** (privacidad).

**2.7 Gestor de Actividades + Outlook Calendar**

* Actividades: tipo (visita/reunión/llamada/demo/otro), asunto, descripción, cuenta/proyecto, contacto(s), responsable, inicio/fin, lugar o link, recordatorio (min), estado ∈ {pendiente, realizada, cancelada}, resultado.
* **Microsoft Graph (Outlook Calendar)**:
  + Botón **“Agregar a Outlook”**: crea/actualiza evento; guardar graph\_event\_id.
  + Participantes: contactos del cliente con email como asistentes.
  + Recordatorios nativos de Outlook (sync).
  + Opcional **pull puntual** desde Outlook para refresh (no polling continuo).
* **Plantillas de actividad** (duración/recordatorio por defecto).
* Vista **Calendario** (día/semana) + **Lista** con filtros.
* **Fallback**: exportar .ics si Graph no está configurado.

**2.8 Checklist por proyecto**

* **Checklist** (colección) por proyecto con **ítems**: título, descripción corta, responsable, prioridad (baja/media/alta), fecha objetivo (opcional), estado ∈ {pendiente, en\_progreso, hecho}, orden, notas, evidencia (adjunto en Drive).
* **Plantillas** reutilizables (ej.: arranque/entrega).
* **Sin dependencias** ni ruta crítica (ligero, foco CRM).

**2.9 Búsqueda Global (FTS5)**

* Indexa: **accounts** (lead/cliente), **contacts**, **proposals**, **projects**, **interactions**, **attachments** (nombre/tags/notas), **activities**, **checklist items**.
* Filtros: estado de cuenta/fase, responsable, tipo de canal, rango de fechas, **incluir eliminados**.

**2.10 Auditoría y Borrado Lógico**

* **Audit log**: quién, qué, cuándo, dónde (host) y versión de app; detallar cambios sensibles (fase, finanzas, permisos, subidas/descargas, sync con Outlook).
* **Soft delete** universal: is\_deleted=1, deleted\_at, deleted\_by. **Nada** se borra físicamente.

**2.11 Configuración (solo Súper Admin)**

* **Drive**: carpeta BD y contenedor de proyectos, credenciales **cifradas**.
* **Outlook/Graph**: client\_id, tenant, redirect URI (PKCE), **scope mínimo** Calendars.ReadWrite.
* Políticas: contraseñas, privacidad (ocultar montos a ciertos roles), **backups** (retención), **anticipos** para campañas (renovación/mantenimiento).
* Plantillas: **mensajes (Hub)**, **actividades**, **checklists**.

**2.12 Actualizaciones e Instalación**

* **PyInstaller → Inno Setup**; datos en %APPDATA%/CRMProyectos (o %PROGRAMDATA%) para preservar config/BD.
* **Auto-update** desde Google Drive:
  + Al iniciar: leer manifest.json en Drive; si remote\_version > local, descargar instalador y actualizar.
  + Validar **hash/firma** antes de aplicar.
* **Migraciones** de BD con **Alembic** al relanzar post-update.

**3) Arquitectura técnica**

* **Python 3.11+, PyQt6 (Qt Widgets/QML),** patrón **MVVM/MVP**.
* **SQLite 3** (WAL, PRAGMA busy\_timeout) + **FTS5**; **SQLAlchemy**; **Alembic**.
* **Google Drive API v3** (OAuth o Service Account); reintentos exponenciales.
* **Microsoft Graph (Outlook)**: OAuth2 **PKCE**; tokens/refresh **cifrados**.
* **Cripto**: cryptography (Fernet/AES-GCM), **Argon2** para contraseñas, **rotación de claves**.
* **Logging**: structlog/logging con rotación; telemetría de errores local (opt-in).
* **Empaquetado**: PyInstaller; Inno Setup (script .iss).
* **SO**: Windows 10/11 (prioritario). (macOS/Linux: roadmap).

**Concurrencia SQLite en Drive (consideraciones):**

* Evitar escrituras largas; **transacciones cortas**; **WAL** habilitado.
* Una sola fuente de **migraciones** (bloqueo a nivel app).
* **Backups** periódicos y **shadow copy** antes de migrar.

**4) Modelo de datos**

Nota: usar **accounts** en lugar de clients (migración inicial si procede).

* users(id, username, email, pwd\_hash, role, is\_active, created\_at, updated\_at, last\_login\_at)
* roles(id, name, permissions\_json) (si se requiere granularidad adicional)
* accounts(id, name, status, industry, size\_band, lead\_source, score, website, notes, converted\_at, converted\_by, is\_deleted)
  + status ∈ {lead, cliente}
* contacts(id, account\_id, name, role\_title, email, phone\_mobile, phone\_work, preferred\_channel, notes, is\_deleted)
* proposals(id, account\_id, title, description, conditions, amount, currency, status, owner\_user\_id, created\_at, updated\_at, is\_deleted)
* projects(id, proposal\_id, account\_id, code, title, description, commercial\_owner\_id, technical\_owner\_id, close\_date, delivery\_date, renewal\_date, base\_doc\_ref, conditions, amount, currency, status, created\_at, updated\_at, is\_deleted)
* project\_interactions(id, account\_id, project\_id, contact\_id, channel, subject, summary, external\_link, attachment\_id, user\_id, occurred\_at, is\_deleted)
  + channel ∈ {whatsapp, teams, meet, email, phone, otro}
* attachments(id, project\_id, proposal\_id, drive\_file\_id, file\_name, ext, size\_bytes, sha256, version, uploaded\_by, uploaded\_at, notes, tags, is\_deleted)
* activities(id, account\_id, project\_id, contact\_id, type, subject, description, location, online\_link, starts\_at, ends\_at, reminder\_minutes, owner\_user\_id, status, outcome, graph\_event\_id, created\_at, updated\_at, is\_deleted)
  + type ∈ {visita, reunion, llamada, demo, otro}; status ∈ {pendiente, realizada, cancelada}
* activity\_templates(id, name, default\_type, default\_duration\_min, default\_reminder\_minutes, default\_description, is\_deleted)
* project\_checklists(id, project\_id, name, created\_by, created\_at, updated\_at, is\_deleted)
* project\_checklist\_items(id, checklist\_id, title, short\_desc, owner\_user\_id, priority, due\_date, status, order\_index, notes, drive\_attachment\_id, created\_at, updated\_at, is\_deleted)
  + priority ∈ {baja, media, alta}; status ∈ {pendiente, en\_progreso, hecho}
* config(id, key, value\_encrypted, updated\_at) (IDs de Drive, credenciales Drive/Graph, políticas, plantillas, anticipos)
* audit\_log(id, timestamp, user\_id, action, entity, entity\_id, details\_json, host, app\_version)

**FTS5** (vistas virtuales + triggers): accounts\_fts, proposals\_fts, projects\_fts, interactions\_fts, attachments\_fts, activities\_fts, checklist\_items\_fts.

**Reglas de negocio clave:**

* Si proposal.status pasa a **Ganado** ⇒ accounts.status = cliente (si era lead) y set converted\_at/by.
* Para **Ganado/Perdido** debe existir **adjunto técnico** asociado.
* Al crear proyecto: si no existe carpeta en Drive, crear y guardar su drive\_folder\_id.

**5) Flujos y criterios de aceptación**

1. **Crear propuesta (Comercial)**
   * Guarda en estado **Borrador**; aparece en listado; audit log registrado; FTS encuentra título/descripción.
2. **Mover de fase (solo Comercial)**
   * No permite **Ganado/Perdido** sin adjunto técnico.
3. **Convertir a proyecto**
   * Crea projects, crea carpeta en Drive, vincula adjuntos; si falla Drive, revierte transacción y muestra error recuperable.
4. **Subir adjunto**
   * Progreso visible; guarda metadatos + drive\_file\_id; si nombre repetido → version += 1; descargable después.
5. **Interacción/Documento**
   * Registro aparece en pestaña y es buscable en ≤ 1 s tras commit (FTS).
6. **Actividades → Outlook**
   * Crear/editar actividad; botón **Agregar a Outlook**: crea/actualiza evento con graph\_event\_id; recordatorio coincide; auditoría registrada; .ics funciona sin Graph.
7. **Checklist de proyecto**
   * Crear checklist/ítems, reordenar, marcar **hecho**; aplicar plantilla; adjuntar evidencia a ítem; auditoría de cambios.
8. **Borrado lógico**
   * is\_deleted=1, deleted\_at/by; oculto por defecto; visible con filtro “incluir eliminados”; auditoría.
9. **Auditoría y reportes**
   * Súper Admin filtra y exporta a CSV/XLSX (incluye cambios de fase, conversiones lead→cliente, subidas/descargas, sync Outlook).
10. **Actualización**

* Si app\_version < remote\_version (en Drive), descargar e instalar; al relanzar, **Alembic** migra sin pérdida; datos/credenciales preservados.

**6) UI/UX**

* **Sidebar**: Dashboard | Leads & Clientes | Oportunidades | Proyectos | Calendario/Actividades | Hub de Mensajería | Buscador | Auditoría | Configuración.
* **Dashboard**: tarjetas (propuestas por estado, proyectos activos, próximas renovaciones y mantenimientos).
* **Leads & Clientes**: pipeline (kanban simple de leads) + grid con filtros; ficha de cuenta con contactos, oportunidades, proyectos históricos y **próximas acciones** (según entrega/renovación).
* **Oportunidades/Proyectos**: listas filtrables; detalle con pestañas: **General | Documentos | Interacciones | Finanzas | Auditoría | Checklist**.
* **Calendario/Actividades**: vista día/semana + lista; botón **Agregar a Outlook**; indicador de sync.
* **Hub de Mensajería**: contactos del cliente, selector de canal, plantillas y deep-link; registro de interacción con resumen y enlace externo.
* **Buscador Global**: barra + filtros, resultados agrupados, resaltado de coincidencias.

**7) Seguridad y cumplimiento**

* Contraseñas **Argon2**; secretos y tokens **AES-GCM (Fernet)**; **rotación de claves**.
* Hashes **SHA-256** de archivos; integridad/verificación.
* **Soft delete** universal; auditoría **exhaustiva**.
* **Backups** programables de la BD a /Backups en la carpeta de Drive con **retención N**.
* **Privacidad**: permitir ocultar montos/finanzas a ciertos roles.

**8) DevEx, QA y DevOps**

**8.1 Estructura del repo**

/app

/ui # PyQt6 (views/widgets)

/core # casos de uso / servicios de dominio

/data # SQLAlchemy, Alembic, FTS

/drive # integración Google Drive

/auth # login, roles, cifrado

/messaging # Hub de Mensajería (deep-links, plantillas)

/activities # Outlook Graph, .ics

/checklists # checklist por proyecto

/campaigns # renovaciones/mantenimientos

/audit # logging y auditoría

/config # pydantic, almacenamiento cifrado

/update # auto-update (manifest.json en Drive)

/tests # PyTest (unitarios, integración, UI)

/installer # PyInstaller + Inno Setup

/docs

**8.2 Calidad**

* Lint/format/type: **Ruff + Black + MyPy**.
* Tests: **PyTest** (unitarios; integración con BD temporal; mocks de Drive/Graph; pytest-qt para UI *smoke* con PyQt6).
* **Cobertura objetivo ≥ 80%** módulos core.

**8.3 CI/CD (GitHub Actions)**

* **CI (PR)**: instalar deps → Ruff/Black/MyPy → PyTest → build PyInstaller (smoke).
* **Release (tag vX.Y.Z)**: build instalador Inno → generar manifest.json (+ sha256) → **subir a Google Drive** (destino configurado) → changelog automático.
* Ramas: main, develop, feat/\*, fix/\*, docs/\*, chore/\*.
* **Conventional Commits + SemVer**; **PRs obligatorios**; **CODEOWNERS**.

**9) Plan de Sprints (8)**

1. Fundaciones: auth, config segura, accounts/leads, contacts, audit base, estructura repo, Alembic inicial.
2. Oportunidades/Propuestas: fases CRM, auditoría, FTS inicial.
3. Proyectos y Drive (carpetas + adjuntos): conversión propuesta→proyecto; metadatos y versionado.
4. Hub de Mensajería + Actividades CRUD/Calendario (esqueleto Graph, .ics).
5. Checklist por proyecto + línea base técnica + plantillas.
6. Outlook Graph robusto (update evento/asistentes, errores transitorios) + campañas (renovación/mantenimiento).
7. Instalador + Auto-update desde Drive + migraciones en producción.
8. Endurecimiento: seguridad, backups, reportes, QA final, manuales y entrega.

**10) Definiciones de “Hecho” (DoD)**

* CRUDs **auditable**; **soft delete** activo; permisos efectivos.
* **Búsqueda global (FTS)** respondiendo correctamente y con filtros.
* Conversión **Lead→Cliente** automática al ganar 1ª propuesta.
* **Ganado/Perdido** exige adjunto técnico; folder de proyecto en Drive.
* **Actividades** sincronizan con **Outlook** (o .ics fallback).
* **Checklist** usable, con plantillas y evidencias.
* **Instalador** operativo; **auto-update** desde Drive; **Alembic** migra sin pérdida de datos.
* Documentación y manuales listos; **cobertura ≥ 80%** en core.

**11) Entregables**

* Documento de arquitectura (alto nivel) y **diagrama ER**.
* Manual de instalación/actualización y **política de backup/recovery**.
* **Matriz de permisos** por rol/operación.
* **README.md** para GitHub; scripts de **Inno Setup** y pipelines de **CI/CD**.
* **Esqueleto de proyecto** (carpetas, dependencias, primeros modelos y migraciones).

**12) Consideraciones importantes**

* **SQLite en Drive**: usar **WAL** y **transacciones cortas**; advertir que mucha concurrencia puede degradar (si escala, evaluar servicio intermedio o Postgres fase 2).
* **Consistencia en Drive**: manejar conflictos de nombre; **versionado lógico**; prefijos de fecha/hora si se requiere.
* **Privacidad**: guardar **metadatos y enlaces** de mensajería, **no** el contenido de conversaciones.