CRM Trinity

Documentación técnica del sistema

Gestión integral de clientes, casos, presupuestos, stock, reportes y backup

Versión del documento: 1.0 | Fecha: 01/06/2026

1. Resumen técnico

CRM Trinity es una aplicación de escritorio orientada a la administración de casos de soporte/servicio técnico, clientes, proveedores, productos, stock, cobros, presupuestos y reportes. El sistema utiliza una interfaz gráfica en Tkinter y persiste la información en PostgreSQL dentro del schema configurable crm.

Área	Detalle
Tipo de aplicación	Aplicación desktop para Windows, desarrollada en Python con Tkinter.
Base de datos	PostgreSQL. Base por defecto: reparaciones. Schema por defecto: crm.
Configuración	Archivo config/app.ini con parámetros de DB, backup, UI y SMTP.
Reportes / PDF	Presupuestos generados con ReportLab en la carpeta salidas.
Correo	Envío SMTP con adjuntos para presupuestos.
Backups	Dump del schema configurado mediante pg_dump, formato custom y compresión.

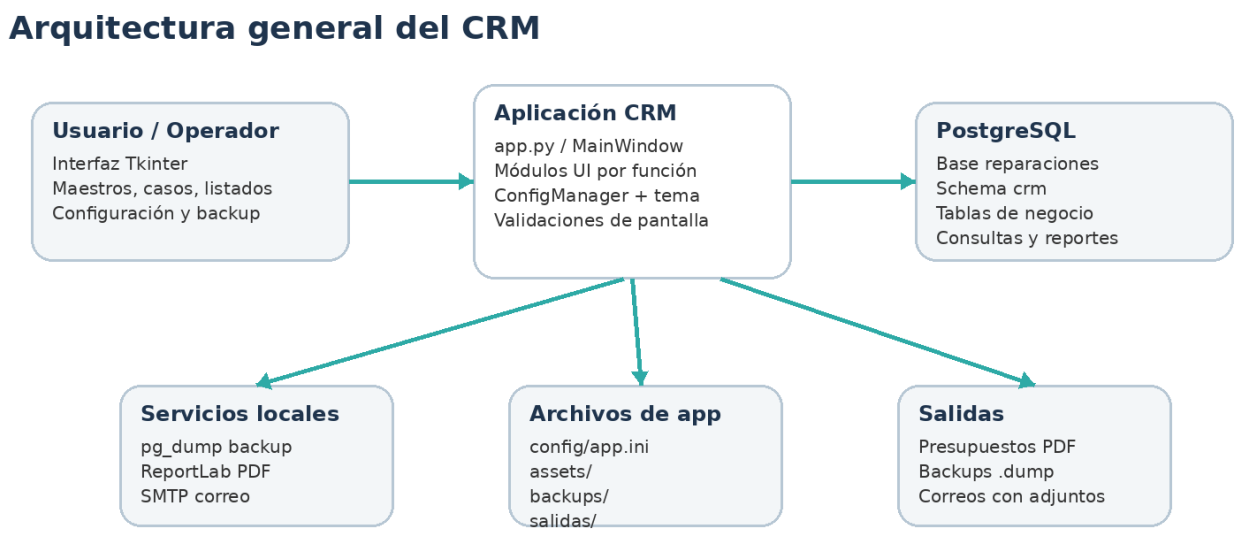
2. Alcance funcional

El sistema centraliza la gestión operativa de un servicio técnico o área de soporte, permitiendo registrar clientes, proveedores, casos, gastos, productos utilizados, estados de cobro y seguimiento posterior.

- Gestión de clientes y proveedores con CUIT, teléfono, email, dirección y notas.
- Alta, edición y baja de casos con motivo, descripción, vencimiento de garantia, gastos, mano de obra y valores a cobrar.
- Plantillas de tipos de caso con descripción base, markup y mano de obra predeterminada.
- Maestro de productos y stock con movimientos de entrada/salida.
- Uso de productos de stock dentro de un caso, con descuento automático del inventario.
- Seguimiento por cliente con notas, fechas de follow-up y estados pendiente/hecho.
- Listados de ventas, caja, rankings, casos pendientes, clientes y proveedores.
- Generación y envío de presupuestos en PDF.
- Backup manual de la base configurada.

3. Arquitectura general

La arquitectura actual es monolítica desktop. La aplicación inicia, válida la base de datos, crea o actualiza el schema necesario, y luego abre la ventana principal. Los módulos de interfaz se cargan bajo demanda para mantener el arranque ordenado.



Componente	Responsabilidad
Capa de interfaz	Ventanas Tkinter/Ttk para panel principal, maestros, detalle de caso, configuración, reportes y seguimiento.
Capa de negocio	Funciónes Python que calculan importes, stock, reportes, presupuestos, backups y envío de correo.
Capa de datos	PostgreSQL con conexión psycopg2, cursores RealDictCursor y search_path al schema configurado.
Archivos locales	Carpetas config, assets, backups y salidas dentro del directorio de la aplicación.

4. Estructura de archivos y módulos

Archivo	Función técnica
app.py	Punto de entrada. Prepara base/schema, maneja errores iniciales y abre la aplicación principal.
db.py	Conexión PostgreSQL, creación de schema/tablas, consultas de negocio, CRUD base, stock y reportes.
main_window.py	Ventana principal, menu general, grilla de casos y acceso a módulos.
case_detail.py	Alta/edición de casos, gastos, stock, presupuesto, mail y borrado.
clientes.py	ABM de clientes con búsqueda y datos de contacto.
proveedores.py	ABM de proveedores con búsqueda y datos de contacto.
productos.py	Maestro de productos, stock actual, ajustes y movimientos.
caso_tipos.py	Plantillas/tipos de caso con defaults de descripción, markup y mano de obra.
cliente_contactos.py	Seguimiento por cliente, notas, follow-ups y próximos pendientes.
backup.py	Resolucion de pg_dump y generación de backups .dump del schema.
budget.py	Generación de presupuesto PDF con ReportLab.
mailer.py	Armado y envío de mensajes SMTP con adjuntos.
config.py	ConfigManager, rutas base, defaults de DB, backup, UI y SMTP.
ui.py / theme.py	Funciónes de icono, tema visual, centrado de ventanas y paleta.
db_config.py / smtp_config.py	Ventanas de configuración de base de datos y correo.

5. Arranque de la aplicación

El flujo de inicio esta pensado para que una instalación nueva pueda crear la base y el schema automaticamente si las credenciales son válidas. Si la conexión falla o la configuración es incompleta, se abre la ventana de configuración de base de datos.

- Instancia ConfigManager y lee config/app.ini.
- Ejecuta ensure_database_exists para crear la base si no existe.
- Ejecuta ensure_schema para crear el schema crm y sus tablas.
- Ejecuta migrate_public_to_schema_if_empty para migrar datos desde public si corresponde.
- Si no hay errores, carga MainWindow y el menu principal.
- Si hay error de conexión, abre DBConfigWindow para corregir host, puerto, usuario, clave, base y schema.

Observacion técnica

La lógica de arranque favorece la recuperación ante primera instalación o credenciales inválidas. Para un instalador final conviene asegurar que config/app.ini se cree en una ruta con permisos de escritura dentro de la carpeta de la aplicación.

6. Configuración del sistema

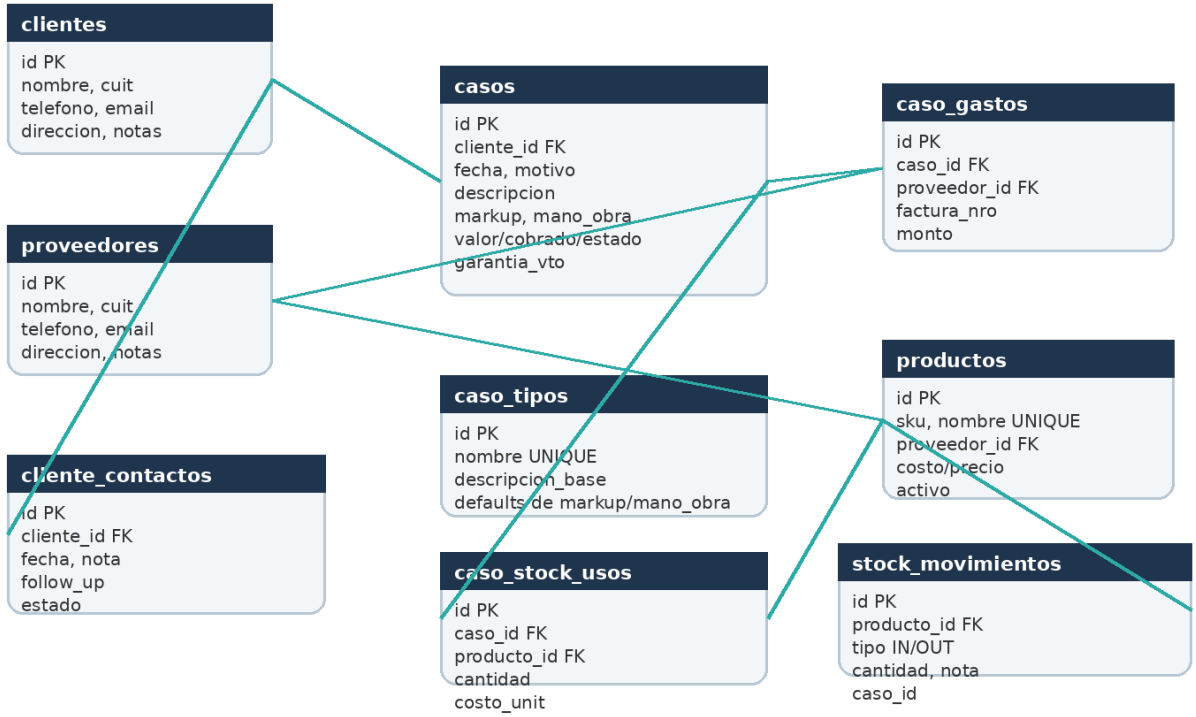
La configuración se administra desde config/app.ini. El archivo se crea automaticamente si no existe y se completan claves faltantes con valores por defecto.

Seccion	Parámetros
[db]	host, port, user, password, dbname, schema_name.
[backup]	pg_dump_path y dest_dir.
[ui]	app_title, icon_path, color_primary, color_secondary, color_bg y color_accent.
[smtp]	host, port, user, password, use_tls, from_name y from_email.

7. Base de datos

El sistema utiliza PostgreSQL como motor de persistencia. La conexión se realiza con psycopg2 y se aplica search_path al schema configurado, normalmente crm, manteniendo public como fallback.

Modelo de datos principal



7.1 Tablas principales

Tabla	Uso
clientes	Datos de clientes: nombre, CUIT, teléfono, email, dirección, notas y fecha de creación.
proveedores	Datos de proveedores: nombre, CUIT, teléfono, email, dirección, notas y fecha de creación.
casos	Cabecera del caso: fecha, cliente, motivo, descripción, importes, estados, pago y garantia.
caso_gastos	Detalle de gastos por proveedor asociados a un caso.
caso_tipos	Plantillas de casos con descripción base, markup y mano de obra predeterminada.
cliente_contactos	Notas de seguimiento por cliente con follow-up y estado.
productos	Maestro de productos con SKU, proveedor, costo, precio y activo.
stock_movimientos	Kardex simple de entradas y salidas de stock.
caso_stock_usos	Productos utilizados en casos, con cantidad y costo unitario.

7.2 Relaciones funcionales

- clientes 1-N casos: un cliente puede tener múltiples casos.
- clientes 1-N cliente_contactos: un cliente puede tener múltiples notas y seguimientos.
- casos 1-N caso_gastos: un caso puede tener varios gastos asociados.
- proveedores 1-N caso_gastos y productos: proveedores se vinculan a gastos y productos.
- productos 1-N stock_movimientos: cada producto tiene historial de movimientos.
- casos N-N productos mediante caso_stock_usos: cada caso puede consumir varios productos y cada producto puede ser usado en distintos casos.

8. Módulos funcionales

8.1 Panel principal

La ventana principal presenta la grilla de casos con fecha, cliente, motivo, descripción breve, gasto total, cobrado, ganancia y estado. Incluye boton de alta rápida y actualización. El doble clic abre el detalle del caso.

- Menu Maestros: Clientes, Proveedores, Tipos de caso y Productos.
- Menu Casos: Alta de caso y seguimiento por cliente.
- Menu Listados: ventas, caja, rankings, casos pendientes y listados generales.
- Menu Configuración: base de datos, SMTP y backup.
- Menu Salir: cierre de aplicación.

8.2 Gestión de casos

El detalle de caso concentra la carga operativa. Permite seleccionar cliente, indicar motivo, descripción, vencimiento de garantia, gastos, productos de stock, markup, mano de obra, valor a cobrar, importe cobrado, estado y medio de pago.

Bloque	Detalle
Datos del caso	Fecha, cliente, motivo, descripción/trabajo realizado y vencimiento de garantia.
Gastos	Detalle de proveedor, factura/referencia y monto. También permite agregar productos desde stock.
Cálculo comercial	Gasto base + markup + mano de obra = valor a cobrar.
Cobro	Importe cobrado, estado del caso, estado de pago y medio de pago.
Acciónes	Guardar, generar presupuesto PDF, enviar por mail, borrar caso y salir.

8.3 Clientes y proveedores

Los maestros de clientes y proveedores tienen estructura similar: búsqueda por nombre, grilla principal y formulario de alta/edición con nombre, CUIT, teléfono, email, dirección y notas.

8.4 Productos y stock

El maestro de productos registra SKU, nombre, descripción, proveedor, costo unitario, precio unitario y estado activo. El stock se calcula a partir de movimientos IN/OUT y se puede consultar el historial por producto.

- Ajuste manual de stock con nota.
- Movimientos tipo IN para entradas y OUT para salidas.
- Uso de productos en casos mediante Agregar desde stock.
- Al guardar un caso se reemplazan los usos de stock del caso y se generan movimientos OUT vinculados.

8.5 Seguimiento por cliente

El módulo de contactos permite registrar notas históricas, fechas de follow-up y estados Pendiente/Hecho por cliente. También permite consultar próximos seguimientos globales hasta una fecha determinada y saltar al cliente seleccionado.

8.6 Listados y reportes operativos

Listado	Contenido
Ventas entre fechas	Detalle por caso y totales de gasto, cobrado y ganancia.
Movimientos de caja	Ingresos, gastos y saldo acumulado por rango de fechas.
Ranking de clientes	Ordenamiento por ganancia o cantidad de ventas.
Casos pendientes	Listado de casos con estado Pendiente.
Listados maestros	Clientes general, clientes con teléfono/mail y proveedores general.

9. Presupuestos PDF y envío de correo

El sistema genera presupuestos en PDF usando ReportLab. El archivo se guarda en la carpeta salidas con nombre Presupuesto_Caso_<ID>.pdf. Desde el detalle del caso se puede generar el PDF o enviarlo por correo como adjunto.

- El presupuesto incluye datos del cliente, datos del caso, descripción, desglose de costos e importe a pagar.
- El envío de correo utiliza SMTP configurado en config/app.ini.
- El módulo mailer.py arma el mensaje, agrega adjuntos existentes y soporta STARTTLS o SMTP_SSL según configuración.
- La pantalla SMTP permite autocompletar Gmail, guardar parámetros y probar el envío.

Dato a revisar antes de producción

El módulo de presupuesto contiene nombre, teléfono y correo de empresa definidos en código. Para una entrega comercial final conviene mover esos datos a config/app.ini o a una pantalla de parámetros de empresa.

10. Backup de base de datos

El backup se ejecuta desde el menu Configuración > Backup de base. La función resuelve la ruta de pg_dump por prioridad: ruta configurada, PATH del sistema o búsqueda en C:/Program Files/PostgreSQL. Luego genera un dump del schema configurado.

Parámetro	Detalle
Herramienta	pg_dump.
Formato	Custom (-Fc), adecuado para restauración selectiva.
Compresión	-Z 9.
Alcance	Solo el schema configurado, por defecto crm.
Nombre archivo	<dbname>_<schema>_<YYYYMMDD_HHMMSS>.dump.
Destino	Parámetro backup.dest_dir, por defecto carpeta backups.

11. Seguridad y consideraciones técnicas

Tema	Recomendación
Credenciales DB/SMTP	Actualmente se guardan en config/app.ini. Para producción se recomienda cifrado local, Windows Credential Manager o archivo protegido por ACL.
Usuarios internos	No se observa módulo de login/roles dentro de la aplicación actual. Puede agregarse si el CRM se instala en equipos compartidos.
Backups	El dump no se cifra por defecto. Si contiene datos sensibles, evaluar cifrado o carpeta protegida.
Correo	Usar contrasenas de aplicación o SMTP transaccional. Evitar cuentas personales sin 2FA.
Integridad de datos	Las FK usan ON DELETE SET NULL o CASCADE según el caso. Revisar comportamiento esperado antes de cargas masivas.
Marca/textos	Hay textos internos con CRM - GTF Soluciones IT. Si el producto final es Trinity, conviene unificar app_title, remitente y presupuesto.

12. Instalación y puesta en marcha

- Instalar PostgreSQL y crear/definir usuario con permisos para crear base y schema, o cargar credenciales de un usuario administrador para la primera ejecución.
- Instalar dependencias Python: psycopg2/psycopg2-binary y reportlab. Tkinter viene incluido en instalaciónes habituales de Python para Windows.
- Configurar config/app.ini o usar la ventana Configuración de base de datos en el primer arranque.
- Ejecutar la aplicación desde app.py o desde el ejecutable empaquetado.
- Ingresar a Configuración > Configurar correo si se utilizaran presupuestos por mail.
- Verificar carpeta salidas para PDFs y carpeta backups para dumps.
- Realizar una prueba de alta de cliente, proveedor, producto, caso, presupuesto, mail y backup.

13. Empaquetado sugerido

Para una distribucion en Windows se puede empaquetar con PyInstaller. La línea exacta puede variar según la estructura final de carpetas, iconos y assets.

```
pyinstaller --noconsole --onefile --name CRM_Trinity app.py --add-data "assets;assets"
```

Nota

Si se usa PostgreSQL externo, no es necesario empaquetar el servidor. Si el instalador debe incluir PostgreSQL, conviene armar un instalador separado o un instalador avanzado con detección de versión.

14. Mantenimiento recomendado

Frecuencia / evento	Acción
Diario / semanal	Verificar backups recientes, revisar errores de correo y controlar espacio en disco.
Mensual	Probar restauración de un dump en una base de prueba.
Antes de actualizar	Copiar config/app.ini, base de datos y carpeta salidas/backups.
Al cambiar marca	Actualizar app_title, iconos, datos de presupuesto y remitente SMTP.
Al agregar tablas	Mantener ensure_schema con migraciones compatibles y documentar nuevas columnas.

15. Mejoras futuras recomendadas

- Módulo de usuarios, perfiles y permisos.
- Cifrado de credenciales sensibles en configuración.
- Válidación de CUIT, emails y teléfonos.
- Exportación de listados a Excel/PDF desde los módulos actuales.
- Dashboard con indicadores: casos pendientes, ventas del mes, cobros pendientes y stock bajo.
- Alertas automáticas de follow-up y vencimientos de garantia.
- Backup programado, rotación de copias y prueba automática de restore.
- Pantalla de parámetros de empresa para logo, teléfono, email, condiciones y textos de presupuesto.
- Registro de auditoria para cambios importantes en casos, cobros y stock.
- Instalador completo con accesos directos, icono, carpeta de datos y permisos adecuados.

16. Checklist de pruebas técnicas

Prueba	Resultado esperado
Arranque inicial	La app crea config/app.ini, base y schema sin errores.
Conexión DB	Probar conexión correcta y error controlado con datos invalidos.
Maestros	Alta/edición/baja de clientes, proveedores, tipos de caso y productos.
Casos	Alta con gastos, edición, eliminación y recálculo de importes.
Stock	Entrada manual, salida por caso y visualización de movimientos.
Presupuesto	Generación de PDF y verificacion visual del archivo.
Correo	Envío de prueba SMTP y envío de presupuesto con adjunto.
Listados	Ventas, caja, ranking, pendientes y listados generales.
Backup	Generar .dump, verificar existencia y realizar restore de prueba.

Documento preparado para Trinity Soluciones IT

Desarrollo, soporte e implementacion de soluciones informaticas para empresas.