

Visão Geral de Cronologia — RC Gestor de Clientes

Este documento resume a evolução do aplicativo, consolidando as principais versões, refatorações e decisões identificadas nos arquivos fornecidos. Ele foi construído a partir de *docs.zip* e *reports.zip* e procura estabelecer uma sequência temporal dos acontecimentos mais relevantes.

Principais Marcos da Linha do Tempo

19 de dezembro de 2024 — Versão v1.4.52 ("Saúde 100% OK")

- **Objetivo:** Auditoria completa de saúde do aplicativo, preparando a base para novas funcionalidades.
- **Melhorias:**
 - Modernização do diálogo de ZIP com um `ProgressDialog` reutilizável e suporte a cancelamento.
 - Correções de type hints usando `TYPE_CHECKING` e `cast`, eliminando erros do **Pyright**.
 - Ajustes de segurança substituindo `try/except/pass` por logs adequados, alinhados com as regras do **Bandit**.
 - Eliminação de busy-wait em uploads, usando um padrão event-driven via `after`, reduzindo o uso de CPU em 5–10%.
 - Thread-safety auditada com todos os riscos médios eliminados; 19/19 testes de upload passando.
- **Documentação:** Relatórios detalhados de auditoria (saúde geral, thread-safety e eliminação de busy-wait) foram gerados, permitindo acompanhar os detalhes técnicos e métricas de qualidade.

20 de dezembro de 2025 — Versão v1.4.72

- **Foco:** Melhoria no sistema de notificações, evolução do módulo **ANVISA** e incremento na cobertura de testes.
- **Novidades:**
 - Notificações ajustadas ao timezone local do usuário, com opção de marcar todas como lidas e coluna "Por" para indicar o autor.
 - Integração com **Winotify** para toasts nativos do Windows e melhorias visuais na UI.
 - Suporte completo ao upload de PDFs no módulo ANVISA, com organização automática em `GERAL/anvisa/{process_slug}/` e interface intuitiva de seleção múltipla.
 - Várias correções menores: padronização de assinatura de funções de upload, tratamento de erros, ajuste do path de upload e validações de tipos.
- **Instalação:** novo instalador `RC-Gestor-Clientes-1.4.72.exe` gerado via PyInstaller e Inno Setup, mantendo compatibilidade com Windows 10+.

26 de dezembro de 2025 — Versão v1.4.93

- **Tema central:** Segurança. A equipe elevou a proteção de credenciais e corrigiu vulnerabilidades críticas.

- **Correções críticas (P0):**

- Removido o arquivo `config/openai_key.txt` — a API key do OpenAI agora é configurada exclusivamente via variável de ambiente (`OPENAI_API_KEY`).
- Corrigida a configuração do PyInstaller para que o `.env` e outras credenciais não sejam incluídos no executável.

- **Melhorias (P1):**

- Migração do armazenamento de tokens Supabase para o **Windows Credential Manager** via biblioteca `keyring`, com migração automática de `auth_session.json` para o cofre.
- Criação de um documento técnico (`docs/SECURITY_MODEL.md`) descrevendo a gestão de chaves Fernet, trade-offs entre segurança e portabilidade, e ordem de precedência das fontes de chave.
- **Documentação atualizada:** README, arquivo `config/README.md` e relatórios em `reports/_qa_codex_tests_smoke_001/` detalham as correções e novos procedimentos.

🕒 28 de dezembro de 2025 — Histórico de Atividades Recentes

- Implementado um pequeno histórico das ações da equipe no Hub, registrando eventos da ANVISA em tempo real.
- Criado `recent_activity_store.py`, um singleton thread-safe que mantém até 200 eventos em memória e notifica observadores quando novos registros são adicionados.
- Interface gráfica adaptada para exibir atividades recentes em uma caixa de texto scrollável com atualização automática, design consistente com o tema do aplicativo.
- Complementado por testes unitários cobrindo adição de eventos, limite de armazenamento, callbacks e formato de saída.

Novembro de 2025 — Decisões de Arquitetura

- **13/11/2025:** o documento **MODULE-MAP v1** descreve a arquitetura em camadas do projeto, mapeando 183 módulos entre UI, Core, Infra, Adapter e outros. Destaca entrypoints, camadas e estatísticas de módulos.
- **10/11/2025:** a **ADR-0001** define a estratégia de timeouts HTTP. Introduz duas variantes (`HTTPX_TIMEOUT_LIGHT` e `HTTPX_TIMEOUT_HEAVY`) para diferenciar operações rápidas (health checks, autenticação) de uploads e downloads pesados, mantendo compatibilidade retroativa via alias.

Refatorações Importantes

- **Janeiro de 2025 (Microfase 3C):** extração dos pollers de notificações, health check e status do `MainWindow` para um componente especializado `MainWindowPollers`, melhorando encapsulamento e testabilidade.
- **Dezembro de 2025 (ANVISA P0.1/P0.2):** centralização de constantes e unificação de utilidades de janela no módulo ANVISA, reduzindo duplicação de código e melhorando a consistência da UI. O documento também aponta dívidas técnicas restantes e áreas ainda acopladas à interface.

Outros Eventos e Patches

- **14/12/2025 — v1.4.37:** patch de correção de travamentos da UI e testes lentos; move o health check para uma thread separada, desativa coverage padrão nos testes e flexibiliza a configuração das chaves do Supabase (`SUPABASE_KEY` / `SUPABASE_ANON_KEY`).
- **18/12/2025:** refatoração da UX de diálogos: diálogos customizados foram substituídos por messageboxes nativas do Windows, enquanto a janela de progresso ZIP foi redesenhada para ser compacta e integrada visualmente à plataforma.

- **26/12/2025:** script SQL adicionando tabelas `org_notifications_user_state` e `org_notifications_hidden` para permitir que usuários escondam notificações individualmente ou de forma global, com políticas RLS para segurança.

Devlogs e Outras Análises

Além dos marcos acima, o diretório `docs_history/` contém mais de 180 devlogs detalhando micro-refatorações, análises de cobertura de testes e verificações específicas (ex.: ordenação de clientes, correção de `.upper()` em `None`, splits do módulo Hub, microfases de clientes e auditoria). Esses arquivos registram eventos distribuídos ao longo de 2024 e 2025, complementando a cronologia automática fornecida no arquivo **analysis_timeline.md**.

Cada devlog normalmente apresenta: - **Data** do evento e tipo de atividade (verificação, refatoração, correção). - **Objetivo** ou problema analisado. - **Mapa de arquivos** inspecionados ou modificados. - **Fluxo de dados** e conclusões sobre o estado do módulo.

Como utilizar estes documentos

- Consulte **analysis_timeline.md** para ver uma linha do tempo detalhada (dia a dia) com todas as menções a datas e resumos extraídos automaticamente dos documentos.
 - Use **cronologia_overview.md** (este arquivo) para ter uma visão de alto nível das versões e principais decisões de arquitetura.
 - Caso identifique lacunas ou datas sem descrição clara, verifique os arquivos originais em `docs`, `reports` ou `docs_history`; outros zips do seu HD podem conter fragmentos adicionais que não foram incluídos aqui.
-