

## POUPE AGORA 2.0 PLANEJAMENTO

```
{
  "meta": {
    "project_name": "Poupe Agora - Plataforma de Assistente Financeiro com IA e WhatsApp",
    "version": "1.0.0",
    "last_update": "2025-11-27",
    "owner": "Marcos Gonçalves",
    "spec_author": "José (Engenheiro de Software - IA Assistant)",
    "description": "Especificação de alto nível da nova arquitetura do Poupe Agora, substituindo o
fluxo atual em n8n por uma aplicação nativa escalável (Node/TS + Next.js + Supabase + Redis
+ OpenAI).",
  },
  "goals": {
    "primary": [
      "Substituir o fluxo atual em n8n por uma arquitetura em código nativo, escalável e
resiliente.",
      "Manter e evoluir os 3 agentes principais: Transações, Consulta/Análises e
Lembretes/Compromissos.",
      "Oferecer interação principal via WhatsApp, com painel web (Admin e Usuário) totalmente
mobile-first.",
      "Suportar escala de centenas a milhares de usuários simultâneos sem travar o
processamento.",
      "Permitir configurar e treinar agentes de IA via painel Admin, sem depender de alterações
diretas no código.",
    ],
    "secondary": [
      "Introduzir diferenciais em relação a concorrentes (GranaZen, Meu Assessor).",
      "Preparar a base para publicação futura como app (APK / PWA).",
      "Manter integrabilidade com gateways externos (PerfectPay/Asaas) via webhooks."
    ],
    "non_goals_for_v1": [
      "Assistente de compras recorrentes com renegociação automática avançada (fica como
feature opcional/futuro).",
      "Integração bancária direta via Open Finance na primeira versão.",
      "Gamificação pesada tipo social network (ranking público entre usuários).",
    ]
  },
  "product_scope": {
    "core_features": [
```

```

    "Registro de transações financeiras via WhatsApp (texto, áudio, imagem de comprovante).",
    "Consulta a saldo, gastos, maiores despesas, resumo por categoria e por período.",
    "Criação e gestão de lembretes financeiros (contas, vencimentos, compromissos).",
    "Painel do usuário com visão de dashboard financeiro, metas e histórico.",
    "Painel Admin com controle de usuários, planos, agentes de IA e integrações.",
    "Integração com PerfectPay/Asaas para criação/atualização de assinaturas.",
    "Memória de conversa contextual por intenção (transação, consulta, lembrete).",
  ],
  "differentiators_v1": [
    "IA proativa que envia alertas e insights financeiros sem o usuário precisar perguntar sempre.",
    "Modo Economia (plano automático de redução de gastos por categoria e metas).",
    "Agente Anti-Endividamento com plano de quitação guiado e acompanhamento.",
    "Planejamento financeiro por objetivos (médio/longo prazo) com micro-metas semanais.",
    "Leitura de documentos financeiros (imagem/PDF de boletos, comprovantes, contratos) usando visão.",
    "Score Financeiro POP (nota e medalhas com base em comportamento financeiro).",
    "Rotinas personalizadas (automations) configuráveis pelo usuário para alertas e ações automáticas.",
  ],
  "optional_future_features": [
    "Monitor de assinaturas recorrentes com renegociação automática (assistente de compras recorrentes).",
    "Integração Open Finance para leitura automática de extratos bancários.",
    "Recomendações de produtos financeiros de terceiros (marketplace).",
  ]
},
"architecture": {
  "style": "Modular Monolith orientado a serviços, com possibilidade de evolução para microserviços.",
  "frontend": {
    "stack": "Next.js (React) + TypeScript + TailwindCSS + shadcn/ui",
    "apps": [
      {
        "name": "Admin Panel",
        "description": "Painel de gestão do sistema, agentes, usuários, planos e integrações.",
        "audience": "Admin / Suporte / Operação"
      },
      {
        "name": "User Panel",
        "description": "Painel do cliente final para visão de finanças, metas, lembretes e automações.",
        "audience": "Usuários finais (individuais e família)."
      }
    ]
  }
}

```

```

],
"requirements": [
  "Design 100% mobile-first.",
  "Pronto para virar PWA / ser encapsulado em APK.",
  "Autenticação compartilhada (SSO) entre Admin e User com controle de roles.",
  "Atualização em tempo quase real das informações (uso de SWR/React Query ou similar).",
]
},
"backend": {
  "stack": {
    "language": "TypeScript",
    "framework": "NestJS ou Fastify/Express com camadas bem definidas (Controllers, Services, Repositories).",
    "db": "Postgres (via Supabase).",
    "cache_queue": "Redis (cache, sessão e fila de jobs).",
    "workers": "BullMQ ou equivalente para processamento assíncrono.",
    "ai": "OpenAI SDK (GPT-4o / GPT-4o-mini, visão, áudio).",
  },
  "modules": [
    "Auth & Users",
    "Billing & Subscriptions",
    "WhatsApp Gateway",
    "Agent Orchestrator (Orquestrador de Agentes de IA)",
    "Finance Core (Transações, Categorias, Metas, Score)",
    "Reminders & Agenda",
    "Automations Engine (Rotinas Personalizadas)",
    "Admin Config (Agentes, Prompts, Integrações)",
    "Observability & Logging"
  ]
},
"integrations": {
  "whatsapp": {
    "current_provider": "Evolution API",
    "future_provider": "API Oficial WhatsApp Business (meta)",
    "strategy": "Criar uma camada de 'WhatsAppService' com interface única para envio/recebimento, isolando o provedor."
  },
  "payments": {
    "providers": [
      "PerfectPay",
      "Asaas"
    ]
  },
},

```

```
    "strategy": "Receber webhooks, normalizar eventos e alimentar o módulo Billing & Subscriptions."
```

```
  },
```

```
  "email": {
```

```
    "provider_example": "Gmail API ou serviço SMTP/Transactional (futuro)."
```

```
  }
```

```
}
```

```
},
```

```
"services": [
```

```
{
```

```
  "key": "auth_users",
```

```
  "name": "Auth & Users",
```

```
  "responsibilities": [
```

```
    "Cadastro e autenticação de usuários.",
```

```
    "Gestão de perfis (nome, email, telefone, whatsapp, plano, status).",
```

```
    "Mapeamento usuário ↔ número de WhatsApp.",
```

```
    "Suporte a contas familiares (owner + membros)."
```

```
  ],
```

```
  "main_entities": [
```

```
    "users",
```

```
    "profiles",
```

```
    "family_members"
```

```
  ],
```

```
  "main_endpoints_examples": [
```

```
    "POST /auth/signup",
```

```
    "POST /auth/login",
```

```
    "GET /me",
```

```
    "GET /users/:id (admin only)"
```

```
  ],
```

```
  "dependencies": [
```

```
    "Supabase (auth/db)"
```

```
  ]
```

```
},
```

```
{
```

```
  "key": "billing",
```

```
  "name": "Billing & Subscriptions",
```

```
  "responsibilities": [
```

```
    "Receber webhooks de pagamento/assinatura.",
```

```
    "Criar/atualizar assinaturas na base interna.",
```

```
    "Controlar status de acesso (ativo, trial, inadimplente, cancelado).",
```

```
    "Definir limites por plano (rotinas, membros, IA calls, etc.)."
```

```
  ],
```

```
  "main_entities": [
```

```
    "subscriptions",
```

```

    "plans",
    "billing_events"
  ],
  "main_endpoints_examples": [
    "POST /billing/webhook/perfectpay",
    "POST /billing/webhook/asaas",
    "GET /billing/subscription-info (auth)"
  ],
  "dependencies": [
    "Auth & Users",
    "Supabase"
  ]
},
{
  "key": "whatsapp_gateway",
  "name": "WhatsApp Gateway",
  "responsibilities": [
    "Receber webhooks do provedor de WhatsApp (Evolution).",
    "Normalizar as mensagens (texto, áudio, imagem, documento).",
    "Enfileirar mensagens em jobs para processamento assíncrono.",
    "Enviar respostas formatadas de volta ao usuário via WhatsApp."
  ],
  "main_entities": [
    "whatsapp_events_log"
  ],
  "queue_jobs": [
    "incoming_message",
    "send_message"
  ],
  "dependencies": [
    "Agent Orchestrator",
    "Redis (fila)",
    "Auth & Users"
  ]
},
{
  "key": "agent_orchestrator",
  "name": "Orquestrador de Agentes de IA",
  "responsibilities": [
    "Identificar usuário e validar permissão/plano.",
    "Gerenciar intenção em aberto (via Redis).",
    "Classificar intenção quando necessário (transacaoFinanceira, consulta, lembrete, etc.).",
    "Direcionar para o agente correto (Transações, Consulta, Lembretes, etc.).",
    "Orquestrar ferramentas (supabase, cálculos, visão, áudio).",

```

"Retornar resposta final formatada e ações a serem executadas (gravar transação, lembrete, meta, etc.)."

```
],
"main_entities": [
  "ai_sessions",
  "ai_agents",
  "ai_agent_prompts",
  "ai_logs"
],
"dependencies": [
  "Finance Core",
  "Reminders & Agenda",
  "Automations Engine",
  "Supabase",
  "Redis",
  "OpenAI"
]
},
{
  "key": "finance_core",
  "name": "Finance Core",
  "responsibilities": [
    "CRUD de transações (despesas, receitas, transferências).",
    "Gestão de categorias e subcategorias.",
    "Cálculo de saldo, totais por período, totais por categoria.",
    "Gestão de metas (objetivos financeiros de curto, médio e longo prazo).",
    "Cálculo do Score Financeiro POP.",
    "Base para modo economia e agente anti-endividamento."
  ],
  "main_entities": [
    "transactions",
    "categories",
    "accounts",
    "goals",
    "debts",
    "financial_snapshots",
    "financial_score"
  ],
  "dependencies": [
    "Auth & Users",
    "Supabase"
  ]
}
```

```

    "key": "reminders",
    "name": "Reminders & Agenda",
    "responsibilities": [
        "Gestão de lembretes financeiros (data, descrição, valor opcional).",
        "Agenda de vencimentos e eventos importantes.",
        "Envio de notificações automáticas (WhatsApp/Email) via jobs agendados."
    ],
    "main_entities": [
        "reminders",
        "scheduled_jobs"
    ],
    "dependencies": [
        "WhatsApp Gateway",
        "Supabase",
        "Redis (scheduler/queue)"
    ]
},
{
    "key": "automations",
    "name": "Automations Engine (Rotinas Personalizadas)",
    "responsibilities": [
        "Permitir que o usuário crie regras do tipo evento → condição → ação.",
        "Eventos exemplo: nova transação, fechamento semanal, gasto acima de limite, etc.",
        "Ações exemplo: enviar mensagem, criar meta, criar lembrete, gerar resumo.",
        "Executar rotinas em background de forma escalável."
    ],
    "main_entities": [
        "user_automations",
        "automation_logs"
    ],
    "dependencies": [
        "Finance Core",
        "Reminders",
        "WhatsApp Gateway",
        "Redis (jobs)"
    ]
},
{
    "key": "admin_config",
    "name": "Admin Config & Management",
    "responsibilities": [
        "Gestão de agentes de IA (prompts, modelos, temperatura, ferramentas).",
        "Gestão de planos, limites, preços (se necessário).",
        "Gestão de integrações (tokens, webhooks, settings).",

```

```

    "Visualização de logs de atendimento e erros."
  ],
  "main_entities": [
    "ai_agents",
    "ai_agent_prompts",
    "admin_users",
    "integration_settings"
  ],
  "dependencies": [
    "Auth & Users",
    "Agent Orchestrator",
    "Observability"
  ]
},
{
  "key": "observability",
  "name": "Observability & Logging",
  "responsibilities": [
    "Registrar logs de IA (prompt, resposta, tempo, tools usadas).",
    "Registrar logs de webhooks e integrações externas.",
    "Expôr métricas para futuros dashboards de monitoramento."
  ],
  "main_entities": [
    "ai_logs",
    "webhook_logs",
    "system_metrics"
  ]
}
],
"ia_agents": [
  {
    "key": "agent_transactions",
    "name": "Agente de Transações Financeiras",
    "primary_intents": [
      "transacaoFinanceira"
    ],
    "input_types": [
      "texto",
      "áudio (transcrição)",
      "imagem de comprovante/nota/recibo"
    ],
    "tools": [
      "CRUD transacoes (Supabase)",
      "consultaCategorias",

```



```

    "addCategoria",
    "infoTransacoes (lista para edição/remover)",
    "limparIntencaoTransacao",
    "calculadora"
  ],
  "memory_strategy": "Sessão por usuário + contexto curto por intenção (chave whatsapp_transacao).",
  "core_behaviour_summary": "Extrair valor, tipo (despesa/receita), categoria, data, descrição amigável; gravar/editar/deletar transações; responder em linguagem natural com resumo formatado.",
  "admin_configurable_fields": [
    "prompt de sistema",
    "exemplos de mensagens",
    "modelo (gpt-4o / gpt-4o-mini)",
    "temperatura",
    "limite de tokens",
    "tools habilitadas"
  ]
},
{
  "key": "agent_consulta",
  "name": "Agente de Consultas/Análises Financeiras",
  "primary_intents": [
    "consulta"
  ],
  "input_types": [
    "texto",
    "áudio (transcrito)"
  ],
  "tools": [
    "transacoes (histórico completo)",
    "lembretes (consultar lembretes)",
    "consultaCategorias",
    "calculadora",
    "apagaIntencaoConsulta"
  ],
  "core_behaviour_summary": "Responder perguntas sobre saldo, totais por categoria, maiores gastos, comportamento de despesas, tendências e análises em linguagem clara, sempre apoiando-se nos dados reais do usuário.",
  "admin_configurable_fields": [
    "prompt de sistema orientado a análises",
    "lista de tipos de análise (saldo, top categorias, variação mês a mês, etc.)"
  ]
},

```

```

{
  "key": "agent_reminders",
  "name": "Agente de Lembretes/Compromissos",
  "primary_intents": [
    "lembrete"
  ],
  "input_types": [
    "texto",
    "áudio (transcrito)"
  ],
  "tools": [
    "salvaLembrete",
    "consultaLembretes",
    "apagaIntencao"
  ],
  "core_behaviour_summary": "Extrair descrição, data (yyyy/mm/dd), valor (se houver) e salvar lembretes financeiros. Perguntar o que faltar. Confirmar criação e mostrar resumo amigável.",
  "admin_configurable_fields": [
    "prompt de sistema",
    "estratégia de validação de data",
    "mensagens padrão em caso de dúvida"
  ]
},
{
  "key": "agent_proactive",
  "name": "Agente Proativo de Alertas & Score",
  "primary_intents": [
    "sistema_proativo"
  ],
  "trigger_mode": "Jobs agendados (não depende de mensagem do usuário).",
  "tools": [
    "Finance Core (totais, metas, metas atrasadas)",
    "Reminders",
    "Automations Engine"
  ],
  "core_behaviour_summary": "Baseado em dados financeiros e metas, enviar alertas: gasto excessivo, meta atrasada, resumo semanal, progresso de dívidas, atualização de Score POP.",
  "admin_configurable_fields": [
    "tipos de alertas ativos",
    "frequência de notificações",
    "modelos de mensagem"
  ]
}

```

```

],
"panels": {
  "admin_panel": {
    "modules": [
      "Dashboard geral (uso, IA, faturamento, assinaturas).",
      "Gestão de usuários e planos (CRUD, upgrade/downgrade, bloqueio).",
      "Gestão de agentes de IA (prompts, modelos, histórico de versões).",
      "Gestão de integrações (tokens, webhooks, status).",
      "Visualização de logs de atendimentos (mensagens, respostas, erros).",
      "Configuração de limites por plano (rotinas, IA calls, membros família).",
    ]
  },
  "user_panel": {
    "modules": [
      "Dashboard financeiro (saldo, gráficos, maiores gastos, resumo rápido).",
      "Transações (lista, filtros, edição manual, vínculos com categorias).",
      "Metas e objetivos (cadastro, progresso, micro-metas).",
      "Família (membros, permissões e visão por pessoa).",
      "Lembretes e agenda (calendário, próximos vencimentos).",
      "Rotinas personalizadas (criação, ativação/desativação, histórico).",
      "Perfil e notificações (dados pessoais, canais, preferências).",
    ]
  }
},
"roadmap": {
  "phase_1": {
    "name": "Fundamentos e Infraestrutura Core",
    "focus": [
      "Configuração de projeto backend (Node/TS, framework, conexão com Supabase e Redis).",
      "Definição e criação do schema inicial (users, profiles, subscriptions, transactions, categories, reminders).",
      "Implementação básica do WhatsApp Gateway (receber/mandar texto).",
      "Implementar Auth & Users + Billing (webhooks + ativação de plano).",
    ]
  },
  "phase_2": {
    "name": "Recriar Fluxo Atual do n8n em Código Nativo",
    "focus": [
      "Implementar Orquestrador de Agentes.",
      "Implementar Agente de Transações, Consulta e Lembretes com o mesmo comportamento atual.",
      "Implementar pipeline de áudio (transcrição) e imagem (comprovantes).",
      "Substituir 100% do fluxo n8n por código nativo."
    ]
  }
}

```

```
]
},
"phase_3": {
  "name": "Painéis Web (Admin + Usuário) - MVP",
  "focus": [
    "Criar painel Admin com gestão básica de usuários, planos e logs.",
    "Criar painel do usuário com dashboard inicial e listagem de transações.",
    "Garantir design mobile-first e pronto para PWA."
  ]
},
"phase_4": {
  "name": "Diferenciais POP & Automations",
  "focus": [
    "Implementar Score Financeiro POP.",
    "Implementar Modo Economia.",
    "Implementar Agente Anti-Endividamento.",
    "Implementar engine de rotinas personalizadas.",
    "Adicionar IA proativa (resumos semanais, alertas automáticos)."
  ]
},
"phase_5": {
  "name": "Escala, Observabilidade e Otimizações",
  "focus": [
    "Adicionar camada forte de observabilidade (logs, métricas, tracing).",
    "Testes de carga e otimizações de performance.",
    "Preparar PWA/APK.",
    "Planejar integração com API oficial do WhatsApp."
  ]
}
}
```