

DOCUMENTO ESTRATÉGICO FUNDACIONAL

# PLATAFORMA DE SAÚDE MENTAL

Ecossistema completo para gestão clínica,  
educação continuada e conexão terapêutica

Versão 2.0

Confidencial

Schema Supabase v17.6

PostgreSQL 17

2025

## SUMÁRIO

---

### ESTRATÉGIA

<b>01</b>	Visão Geral do Projeto	3
<b>02</b>	Origem e Motivação	4
<b>03</b>	O Problema que Resolvemos	5
<b>04</b>	O Mercado e seus Concorrentes	6
<b>05</b>	Lacunas de Mercado Identificadas	7

### PRODUTO

<b>06</b>	Arquitetura do Produto	8
<b>07</b>	O Sistema Multi-Papel e Multi-Tenant	9
<b>08</b>	Módulos do Sistema de Gestão	11
<b>09</b>	Plataforma de Microlearning	13
<b>10</b>	Modelo de Negócio e Monetização	15

### BANCO DE DADOS

<b>11</b>	Modelo de Dados — Schema Real	17
<b>12</b>	Enums e Tipos — Referência Completa	22
<b>13</b>	Funções RPC — Lógica de Negócio no Banco	24

### EXECUÇÃO

<b>14</b>	Estratégia de Fases de Desenvolvimento	26
<b>15</b>	Cronograma do MVP (3 Meses)	28
<b>16</b>	Riscos e Mitigações	29
<b>17</b>	Estratégia de Adoção e Crescimento	30
<b>18</b>	Visão de Longo Prazo	31

01

# Visão Geral do Projeto

Este documento descreve de forma completa e minuciosa o projeto de uma **Plataforma de Saúde Mental** — um ecossistema digital integrado que reúne gestão clínica, educação continuada e conexão entre profissionais e pacientes em um único ambiente.

A versão 2.0 deste documento incorpora a realidade do banco de dados já construído: tabelas reais, enums implementados, funções RPC em produção e decisões arquiteturais tomadas durante o desenvolvimento. Este não é mais um documento de intenções — é o espelho fiel do que existe e do que está sendo construído.

*"Não estamos construindo um software de gestão. Estamos construindo o ecossistema que o profissional de saúde mental brasileiro nunca teve."*

**O que mudou na v2.0:** Adicionados os capítulos 11, 12 e 13 com a documentação completa do schema real do banco de dados (Supabase/PostgreSQL 17.6), incluindo as 60+ tabelas do schema público, todos os enums implementados e as principais funções RPC. O capítulo 07 foi expandido com os tipos de tenant reais. O capítulo 10 foi atualizado com as chaves de plano reais (plans.target: patient, therapist, clinic, supervisor).

## Pilares Fundadores

<b>Confiança Clínica</b> Seriedade, conformidade com LGPD e segurança de dados sensíveis desde a fundação.	<b>Redução de Fricção</b> Cada funcionalidade deve economizar tempo real do profissional, nunca criar mais trabalho.
<b>Efeito de Rede</b> Quanto mais profissionais entram, mais o produto vale para cada um deles.	<b>Educação como Aquisição</b> Conteúdo de microlearning que atrai, engaja e converte o profissional ideal.
<b>RLS por Design</b> Row Level Security garante que cada tenant só acessa seus próprios dados — sem vazamento cruzado.	<b>Multi-Tenant Real</b> A coluna tenant_id está em todas as tabelas críticas, garantindo isolamento total.

02

## Origem e Motivação

O projeto surgiu da observação direta de um problema recorrente: profissionais de saúde mental altamente capacitados clinicamente, mas completamente despreparados para gerir o aspecto administrativo e financeiro de suas práticas.

A pandemia de COVID-19 acelerou a digitalização do setor — o atendimento online normalizou-se, a demanda por terapia explodiu e o número de profissionais cresceu. Porém, as ferramentas disponíveis no mercado não acompanharam essa evolução em profundidade. Resolvem o agendamento, mas ignoram a clínica. Resolvem o financeiro, mas ignoram o paciente.

A motivação central é construir o produto que o mercado deveria ter criado há anos: uma plataforma que entende que o terapeuta **também é paciente**, que a clínica é feita de relações humanas, e que educação e prática clínica são inseparáveis.

### O Papel do Parceiro Acadêmico

Um dos diferenciais estratégicos do projeto é a parceria com um profissional em formação em psicologia e já atuante como psicanalista. Essa combinação une a visão clínica prática com o rigor acadêmico, permitindo a produção de conteúdo de microlearning com autoridade real.

Com o tempo, esse parceiro poderá atuar como supervisor clínico dentro da plataforma. O schema já prevê isso: `plans.target = 'supervisor'` e `tenants.kind = 'supervisor'` são tipos nativos do sistema, não adições futuras.

03

## O Problema que Resolvemos

### 1. Fragmentação de Ferramentas

O profissional usa Google Agenda para agendamento, WhatsApp para comunicação, planilha para financeiro, Word para prontuário e e-mail para cobrança. São 5 ferramentas para uma única prática clínica.

### 2. Ausência de Visão Clínica nos Sistemas Atuais

Os softwares existentes tratam o atendimento como uma transação comercial. Não há suporte para acompanhamento de evolução do paciente, aplicação de escalas validadas ou registro estruturado por abordagem.

### 3. Isolamento Profissional

Terapeutas trabalham em silos. Não há plataforma que facilite encaminhamentos entre colegas, supervisão clínica estruturada ou rede de colaboração profissional.

### 4. Falta de Educação em Gestão

A formação em psicologia, psicanálise e psiquiatria é excelente clinicamente, mas praticamente nula em gestão. O profissional não sabe precificar, lidar com inadimplência

ou conhece suas obrigações fiscais e de LGPD.

### **5. Experiência do Paciente Negligenciada**

O paciente é tratado como destinatário passivo do serviço. Não existe ferramenta que engaje o paciente entre sessões, acompanhe seu humor, envie tarefas terapêuticas ou registre seu progresso de forma visual.

04

## O Mercado e seus Concorrentes

O mercado brasileiro de software para saúde mental está em crescimento acelerado, mas ainda dominado por soluções parciais. Os principais players cobrem bem o básico administrativo, porém falham na profundidade clínica e na experiência do usuário.

Player	Pontos Fortes	Lacunas Críticas
PsicoManager	Prontuário, agendamento sólido	UX datada, sem app do paciente, sem microlearning
Clínica Ágil	Interface moderna, financeiro	Sem escalas clínicas, sem rede entre profissionais
Amplimed	Multi-especialidade, robusto	Complexo demais para terapeuta solo, custo elevado
Nuvem Psicologia	Foco em psicólogos, simples	Funcionalidades limitadas, sem crescimento de produto
Google Agenda + Planilha	Gratuito, familiar	Não é um produto — é um remendo

*Nossa plataforma compete em uma categoria nova: **ecossistema terapêutico integrado**. Enquanto os concorrentes vendem software de gestão, vendemos crescimento profissional, conexão clínica e educação continuada — com a gestão como consequência natural.*

05

## Lacunas de Mercado Identificadas

A análise competitiva revelou oportunidades concretas que nenhum player atual preenche de forma satisfatória. São estas lacunas que definem nossa vantagem estratégica.

### Lacuna 1 — Acompanhamento de Progresso Clínico Visual

Nenhum sistema entrega um painel visual que mostre a evolução do paciente sessão a sessão. Nossa tabela agenda\_eventos já registra cada sessão com status, modalidade e preço — base para dashboards longitudinais.

### Lacuna 2 — Ferramentas para o Paciente entre Sessões

Um módulo do paciente com diário de humor, tarefas e check-ins é praticamente inexistente no Brasil. A tabela patients já prevê user\_id, permitindo que pacientes tenham login nativo na plataforma.

### Lacuna 3 — Escalas Psicológicas Integradas

PHQ-9, GAD-7, BDI e outras escalas não estão integradas com envio ao paciente e correção automática. O sistema de determined\_commitments com campos dinâmicos forma a base técnica para isso.

#### **Lacuna 4 — Rede de Encaminhamentos**

Terapeutas precisam encaminhar pacientes para colegas. O campo `patients.encaminhado_por` e o status "Encaminhado" já existem na tabela de pacientes. A estrutura de rede é o passo seguinte.

#### **Lacuna 5 — Módulo de Supervisão Clínica**

Nenhum sistema tem módulo específico para supervisores. No nosso schema: `plans.target = "supervisor"`, `tenants.kind = "supervisor"` e `plans.max_supervisees` já estão modelados nativamente.

#### **Lacuna 6 — Educação em Gestão para a Área**

A plataforma de microlearning — produto 2 — é um diferencial único com conteúdos específicos sobre gestão de clínica, precificação, LGPD aplicada à psicologia e finanças do autônomo.

#### **Lacuna 7 — IA de Suporte Clínico**

Sugestão de bibliografias por apresentação clínica e rascunho de evoluções por voz. O campo `agenda_eventos.extra_fields (jsonb)` e os campos de notas estruturadas já preparam o terreno para IA.

06

## Arquitetura do Produto

A plataforma é composta por dois produtos complementares que compartilham a mesma base de usuários, infraestrutura Supabase e identidade de marca:

PRODUTO 1

Sistema de Gestão Clínica

Agendamento com r  
com Pix/Asaas, mult  
paciente, recibo de pro

### Stack Tecnológica Real

Componente	Descrição
Supabase (Backend)	PostgreSQL 17.6, Auth, Storage, RLS, Realtime. Hospeda 60+ tabelas do schema público.
Auth Supabase	JWT com refresh token. <code>auth.uid()</code> em todas as RLS policies. Login por e-mail e Google.
Asaas (Pagamentos)	Integração via webhooks para Pix, boleto. Tabela <code>payment_settings</code> guarda configurações por owner.
Sistema de Notificações	<code>notification_queue</code> , canais (email, WhatsApp, SMS), opt-in por paciente e cleanup automático.

### Integração entre os Dois Produtos

- O mesmo login (`auth.users` → `profiles`) acessa os dois produtos — conta única.
- Banners rotativos no dashboard exibem conteúdos do microlearning (tabela `login_carousel_slides` já existe no schema).
- O modelo de planos (`plans`) inclui acesso ao microlearning como benefício por meio de `plan_features`.
- O sistema de módulos (`modules`, `tenant_modules`) controla quais funcionalidades cada `tenant` tem acesso.
- O conteúdo de gestão de clínica educa o profissional para usar melhor as funcionalidades do próprio sistema.



07

## O Sistema Multi-Papel e Multi-Tenant

Um dos diferenciais arquiteturais mais importantes da plataforma é o sistema de **multi-papel por usuário combinado com multi-tenant**. Na maioria dos concorrentes, um usuário é ou terapeuta ou paciente. Em nossa plataforma, um único usuário pode exercer múltiplos papéis simultaneamente, refletindo a realidade clínica.

*Insight central: O terapeuta também faz análise pessoal. Logo, ele é simultaneamente profissional de saúde mental e paciente de outro profissional. O sistema precisa respeitar e suportar essa dualidade de forma natural e sem fricção.*

### Os Papéis do Sistema

PAPEL 01	Paciente	Terapeuta
PAPEL 03	Clínica	Supervisor

### Tipos de Tenant — `tenants.kind`

Chave (kind)	Tipo	Descrição
therapist	Terapeuta Solo	Terapeuta autônomo com workspace próprio. Um tenant por profissional.
clinic_coworking	Clínica Coworking	Múltiplos terapeutas com espaço compartilhado, agenda independente por profissional.
clinic_reception	Clínica com Recepção	Recepcionista centralizada gerenciando agendamento de múltiplos profissionais.
clinic_full	Clínica Completa	Gestão completa: recepção, financeiro centralizado, repasse, relatórios gerenciais.
supervisor	Workspace de Supervisão	Ambiente dedicado à supervisão clínica com acesso controlado a casos.

### Regras de Transição entre Papéis

- Todo usuário começa com `profiles.account_type = 'free'` — sem perfil definido ainda.
- O usuário escolhe ser paciente ('patient') ou terapeuta ('therapist') — o campo é imutável após a escolha.

- Um Terapeuta pode criar ou se vincular a uma Clínica via `tenant_members` (role: `tenant_admin` ou `therapist`).
- Um usuário Terapeuta pode simultaneamente ser Paciente de outro terapeuta — RLS garante segregação total.
- Papéis globais de plataforma (ex: editor de microlearning) ficam em `profiles.platform_roles[]`, independentes de tenant.

08

## Módulos do Sistema de Gestão

M1

### Autenticação e Multi-Papel

Fundação de toda a plataforma. Usa `auth.users` (Supabase Auth) + `profiles` para dados do usuário. O `profiles.role` controla acesso administrativo; `profiles.account_type` controla o tipo de conta.

- Login com e-mail, Google e futuramente Apple
- JWT com `auth.uid()` usado em todas as RLS policies
- Middleware de permissão por papel em todas as rotas
- Auditoria via `support_sessions` — admins SaaS com token temporário
- `profiles.platform_roles[]` para permissões globais independentes de tenant

M2

### Agendamento com Recorrência

Sistema de agenda completo com recorrência, exceções, bloqueios e agendamento online pelo paciente.

- `agenda_regras_semanais` — disponibilidade semanal (dia, horário, modalidade)
- `agenda_eventos` — cada sessão individual com status: agendado, realizado, faltou, cancelado, remarcar
- `recurrence_rules` — regras de recorrência (weekly, biweekly, monthly, yearly, custom\_weekdays)
- `recurrence_exceptions` — exceções pontuais: cancelamento, remarcação, feriado
- `agendador_solicitacoes` — agendamento online pelos pacientes com aprovação do terapeuta

M3

### Cadastro de Pacientes

Cadastro completo do paciente com 40+ campos. Suporta `patient_scope = 'clinic'` e `'therapist'`.

- `patients` — dados pessoais, endereço completo, responsável, CPF/RG, notas internas
- Status: Ativo | Inativo | Alta | Encaminhado | Arquivado
- `patient_invites` — link de auto-cadastro com token, expiração e limite de usos
- `patient_intake_requests` — ficha de anamnese via link público (`create_patient_intake_request_v2`)
- `patient_tags` e `patient_groups` — organização e segmentação de pacientes

M4

### Financeiro

Controle financeiro completo com receitas, despesas, parcelamentos e repasse para clínica.

- `financial_records` — receita/despesa com status: pending, paid, partial, overdue, cancelled, refunded
- Criação automática via trigger `auto_create_financial_record_from_session` quando sessão é realizada
- `billing_contracts` — contratos de cobrança por paciente (preço, frequência, método)

- therapist\_payouts — repasse financeiro para terapeutas em clínicas
- insurance\_plans e insurance\_plan\_services — gestão de convênios e planos de saúde

**M5**

**Notificações**

Sistema de notificações multi-canal com fila, agendamento, opt-in por canal e limpeza automática.

- notification\_queue — fila (pendente → processando → enviado)
- Canais: email, WhatsApp, SMS — opt-in individual por paciente em notification\_preferences
- notification\_templates e email\_templates\_tenant — templates personalizáveis por tenant
- Cancelamento automático quando sessão é cancelada ou paciente faz opt-out

**M6**

**Serviços e Precificação**

Catálogo de serviços por terapeuta/clínica com vínculo à recorrência e precificação por sessão.

- services — catálogo de serviços (consulta, avaliação, retorno etc.) por owner/tenant
- recurrence\_rule\_services — serviços vinculados a uma recorrência com preço e desconto
- Prioridade: recurrence\_rule\_services > recurrence\_rules.price > agenda\_eventos.price

09

## Plataforma de Microlearning

O produto de microlearning é o motor de aquisição orgânica da plataforma. Profissionais chegam pelo conteúdo e descobrem o sistema de gestão. É a estratégia que transforma educação em produto.

*Insight-chave: o melhor canal de aquisição de um terapeuta é outro terapeuta. O microlearning cria o contexto social onde isso acontece naturalmente.*

### Trilhas de Conteúdo por Especialidade

#### Gestão de Clínica para Psicólogos (TCC, Gestalt, Humanista)

- Como precificar sua sessão sem culpa e com consciência de mercado
- LGPD na prática clínica: o que você precisa fazer hoje
- Gestão de inadimplência sem comprometer o vínculo terapêutico
- Prontuário eletrônico: o que é obrigatório pelo CFP
- Organização financeira para o profissional autônomo

#### Gestão de Clínica para Psicanalistas

- O setting e o dinheiro: como falar sobre honorários sem quebrar a transferência
- Frequência de sessões e modelo de cobrança na psicanálise
- Como estruturar uma clínica particular sendo psicanalista
- Supervisão e análise pessoal: como e onde buscar

#### Gestão de Clínica para Psiquiatras

- Receituário, laudos e documentação clínica
- Credenciamento e faturamento com planos de saúde
- Integração com outros profissionais de saúde mental
- Como montar um consultório particular de psiquiatria

#### Para Clínicas Multiprofissionais

- Como estruturar o repasse financeiro entre profissionais
- Gestão de agenda compartilhada sem conflitos
- Contrato de parceria entre clínica e profissional

10

## Modelo de Negócio e Monetização

O modelo de negócio é SaaS com camada freemium. Os planos são gerenciados pela tabela **plans** com preços em **plan\_prices** e vitrine pública em **plan\_public**. Cada plano tem um **target** que define para qual tipo de conta ele é válido.

### Estrutura de Planos

Gratuito	Profissional	Clínica	Supervisor	Conteúdo
target: patient / therapist	target: therapist	target: clinic	target: supervisor	target: patient
Até 10 pacientes Agendamento básico Conteúdo gratuito	Pacientes ilimitados Prontuário completo Financeiro + escalas Todas as trilhas	Multi-terapeuta Repasse financeiro Agenda compartilhada Toda a equipe	Limite de supervisionados Acesso controlado a casos Workspace de supervisão	Acesso completo às trilhas Sem sistema de gestão Assinatura mensal

### Infraestrutura de Assinaturas

- **subscriptions** — assinatura ativa do usuário ou tenant (XOR: ou **user\_id** ou **tenant\_id**, nunca ambos)
- Status de assinatura: **pending** | **active** | **past\_due** | **suspended** | **cancelled** | **expired**
- **subscription\_intents\_personal** e **subscription\_intents\_tenant** — intenção de pagamento antes da ativação
- Função **activate\_subscription\_from\_intent()** — ativa assinatura após confirmação de pagamento
- Função **change\_subscription\_plan()** — muda plano com log em **subscription\_events**
- **entitlements\_invalidation** — cache de entitlements é invalidado quando o plano muda
- **plan\_prices** — histórico de preços por plano com intervalos (month/year)

# Modelo de Dados — Schema Real

Documentação minuciosa das tabelas principais do schema **public** do banco de dados PostgreSQL 17.6 via Supabase. Este capítulo é a referência autoritativa para qualquer decisão de desenvolvimento.

*Convenção: Todas as tabelas usam uuid como PK via gen\_random\_uuid(). Campos tenant\_id e owner\_id estão presentes em todas as tabelas de negócio. PK = Primary Key | FK = Foreign Key | NN = Not Null*

## Tabela: profiles

Extensão do auth.users do Supabase. Criado automaticamente no signup. Contém dados públicos do usuário e configurações de perfil.

public.profiles		
id <b>PK</b> <b>FK</b>	uuid	Mesmo ID do auth.users — FK implícita
role	text	Papel no sistema SaaS: saas_admin   tenant_member   portal_user   patient
account_type	text	Tipo de conta: free   patient   therapist   clinic. Imutável após definição.
platform_roles	text [ ]	Papéis globais de plataforma (ex: editor de microlearning). Atribuído pelo saas_admin.
full_name	text	Nome completo do usuário
avatar_url	text	URL pública no Supabase Storage
phone	text	WhatsApp no formato (99) 99999-9999
bio	text	Breve apresentação pública (máx 300 chars no front)
language	text	Default: 'pt-BR'
timezone	text	Default: 'America/Sao_Paulo'
social_custom	jsonb	Redes sociais adicionais: [{name, url}]

## Tabela: tenants

Representa um workspace isolado — pode ser terapeuta solo, clínica ou workspace de supervisão. O kind é imutável após criação.

public.tenants		
id <b>PK</b>	uuid	ID único do tenant
name	text	Nome do workspace (clínica ou nome do terapeuta)

public.tenants		
kind <b>NN</b>	text	therapist   clinic_coworking   clinic_reception   clinic_full   supervisor
created_at	timestampz	Data de criação

## Tabela: patients

Registro completo do paciente. O patient\_scope define se pertence à clínica ou a um terapeuta específico.

public.patients		
id <b>FK</b>	uuid	ID único do paciente
tenant_id <b>FK NN</b>	uuid	Workspace ao qual pertence
owner_id <b>FK</b>	uuid	Quem criou/é responsável pelo paciente
responsible_member_id <b>NN</b>	uuid	Membro responsável no tenant
therapist_member_id	uuid	Terapeuta vinculado (obrigatório quando patient_scope = therapist)
user_id <b>FK</b>	uuid	Conta de login do paciente (quando usuário nativo)
patient_scope	text	'clinic'   'therapist' — define a quem o paciente pertence
status	text	Ativo   Inativo   Alta   Encaminhado   Arquivado
nome_completo <b>NN</b>	text	Nome completo do paciente
cpf	text	CPF sem formatação (validado por regex <code>^d{11}\$</code> )
data_nascimento	date	Data de nascimento
cobranca_no_responsavel	boolean	Cobrar no responsável em vez do paciente
notas_internas	text	Notas privadas do terapeuta (não visíveis ao paciente)



## Tabela: agenda\_eventos

Cada sessão individual na agenda. Gerada manualmente ou automaticamente por regras de recorrência. O campo billed controla se já foi gerada cobrança.

public.agenda_eventos		
id <small>FK</small>	uuid	ID único do evento
tenant_id <small>NN</small>	uuid	Workspace
owner_id <small>NN</small>	uuid	Terapeuta dono do evento
patient_id <small>FK</small>	uuid	Paciente da sessão (null em bloqueios)
tipo	tipo_evento_agenda	'sessao'   'bloqueio'
status	status_evento_agenda	agendado   realizado   faltou   cancelado   remarcar
inicio_em <small>NN</small>	timestampz	Início da sessão (com timezone)
fim_em <small>NN</small>	timestampz	Fim — constraint: fim_em > inicio_em
modalidade	text	'presencial'   'online' — default: 'presencial'
recurrence_id <small>FK</small>	uuid	Regra de recorrência que gerou este evento
price	numeric(10,2)	Valor da sessão em BRL
billed	boolean	true = cobrança já foi gerada para esta sessão
extra_fields	jsonb	Campos customizáveis por abordagem/tenant
insurance_plan_id	uuid	Plano de saúde da sessão (quando convênio)

## Tabela: financial\_records

Registro financeiro central. Gerado automaticamente por trigger quando sessão é marcada como realizada. Suporta parcelamento, split de comissão e múltiplos status.

public.financial_records		
id <small>FK</small>	uuid	ID único
owner_id <small>NN</small>	uuid	Terapeuta dono do registro
tenant_id	uuid	Clínica (quando em contexto de clínica)
type	financial_record_type	'receita'   'despesa'
amount <small>NN</small>	numeric(10,2)	Valor bruto
discount_amount	numeric(10,2)	Desconto aplicado
final_amount	numeric(10,2)	Valor final após desconto
clinic_fee_pct	numeric(5,2)	Percentual da clínica (0–100)

public.financial_records		
net_amount	numeric(10,2)	GENERATED: amount - clinic_fee_amount (valor líquido do terapeuta)
status <small>NN</small>	text	pending   paid   partial   overdue   cancelled   refunded
agenda_evento_id <small>FK</small>	uuid	Sessão que originou a cobrança
installments	smallint	Número de parcelas (default: 1)
installment_group	uuid	UUID que agrupa parcelas de um mesmo parcelamento
deleted_at	timestampz	Soft-delete — null = registro ativo

## Tabelas de Suporte — Referência Rápida

Tabela	Finalidade	Observação
subscriptions	Assinatura ativa do usuário ou tenant	XOR: user_id OU tenant_id
plans	Planos com preço e target	target: patient/therapist/clinic/supervisor
plan_features	Features incluídas em cada plano	Vincula plans ↔ features
modules + tenant_modules	Módulos disponíveis por tenant	Controle granular de acesso
notification_queue	Fila de notificações multi-canal	Email, WhatsApp, SMS
patient_invites	Links de auto-cadastro para pacientes	Token + expiração + limite de usos
patient_intake_requests	Fichas de anamnese via link público	Aprovadas pelo terapeuta
billing_contracts	Contratos de cobrança por paciente	Preço, frequência, método
services	Catálogo de serviços do terapeuta	Substitui professional_pricing (deprecated)
insurance_plans	Convênios e planos de saúde	Vinculado a sessões e recorrências
therapist_payouts	Repasse financeiro do terapeuta	Período, sessões, valor líquido
support_sessions	Sessões de suporte do saas_admin	Token temporário com TTL (1–120 min)
user_settings	Preferências de UI por usuário	Theme, layout, modo de menu
feriados	Feriados nacionais/locais	Integração com bloqueios de agenda

Tabela	Finalidade	Observação
global_notices	Avisos da plataforma	Com dismissal individual por usuário

12

## Enums e Tipos — Referência Completa

O schema define enums PostgreSQL para todos os campos com valores discretos. Isso garante integridade de dados no nível do banco — sem strings inválidas passando pelo sistema.

### Enums do Schema public

<code>status_evento_agenda</code>	<code>recurrence_type</code>	<code>weekly</code> <code>biweekly</code> <code>monthly</code> <code>custom_weekdays</code>
<code>tipo_evento_agenda</code>	<code>recurrence_exception_type</code>	<code>cancel_session</code> <code>reschedule</code> <code>patient_missed</code> <code>therapist_missed</code> <code>holiday_block</code>
<code>financial_record_type</code>	<code>status_agenda_serie</code>	<code>ativo</code> <code>pausado</code> <code>cancelado</code>
<code>tipo_excecao_agenda</code>	<code>determined_field_type</code>	<code>text</code> <code>textarea</code> <code>number</code>

### Constraints de Texto — Equivalentes a Enums

Tabela.Coluna	Valores Permitidos
<code>profiles.role</code>	<code>saas_admin</code>   <code>tenant_member</code>   <code>portal_user</code>   <code>patient</code>
<code>profiles.account_type</code>	<code>free</code>   <code>patient</code>   <code>therapist</code>   <code>clinic</code>
<code>tenants.kind</code>	<code>therapist</code>   <code>clinic_coworking</code>   <code>clinic_reception</code>   <code>clinic_full</code>   <code>supervisor</code>
<code>patients.status</code>	<code>Ativo</code>   <code>Inativo</code>   <code>Alta</code>   <code>Encaminhado</code>   <code>Arquivado</code>
<code>patients.patient_scope</code>	<code>clinic</code>   <code>therapist</code>
<code>financial_records.status</code>	<code>pending</code>   <code>paid</code>   <code>partial</code>   <code>overdue</code>   <code>cancelled</code>   <code>refunded</code>
<code>subscriptions.status</code>	<code>pending</code>   <code>active</code>   <code>past_due</code>   <code>suspended</code>   <code>cancelled</code>   <code>expired</code>
<code>subscriptions.interval</code>	<code>month</code>   <code>year</code>
<code>plans.target</code>	<code>patient</code>   <code>therapist</code>   <code>clinic</code>   <code>supervisor</code>
<code>therapist_payouts.status</code>	<code>pending</code>   <code>paid</code>   <code>cancelled</code>
<code>agenda_eventos.modalidade</code>	<code>presencial</code>   <code>online</code>
<code>user_settings.theme_mode</code>	<code>light</code>   <code>dark</code>

13

## Funções RPC — Lógica de Negócio no Banco

O sistema usa extensivamente **funções PostgreSQL SECURITY DEFINER** para encapsular lógica de negócio complexa e garantir atomicidade. Todas são chamadas via `supabase.rpc()` no frontend.

*Atenção arquitetural: Funções SECURITY DEFINER executam com as permissões do dono (supabase\_admin), não do usuário chamador. Isso permite operações controladas que burlariam o RLS. Use com responsabilidade.*

### Funções Críticas de Negócio

Função	O que faz	Observação crítica
<code>create_clinic_tenant(p_name)</code>	Cria tenant do tipo clinic e vincula usuário atual como tenant_admin	Atômico: cria tenant + tenant_members em uma transação
<code>activate_subscription_from_intent(p_intent_id)</code>	Ativa assinatura após pagamento confirmado (intent.status = paid)	Valida target (clinic/therapist/supervisor). Suporta todos os 3 tipos.
<code>change_subscription_plan(p_subscription_id, p_new_plan_id)</code>	Muda plano de assinatura ativa com log em subscription_events	Valida compatibilidade de target (clinic ≠ therapist)
<code>cancel_subscription(p_subscription_id)</code>	Cancela assinatura e invalida cache de entitlements	Logs em subscription_events, invalida entitlements_invalidation
<code>create_financial_record_for_session(...)</code>	Cria cobrança para sessão com idempotência (não duplica)	Marca billed=true em agenda_eventos após criar
<code>auto_create_financial_record_from_session()</code>	TRIGGER: cria financial_record quando sessão é marcada realizada	Prioridade: recurrence_rule_services > recurrence_rules.price > agenda_eventos.price
<code>create_patient_intake_request_v2(p_token, p_payload)</code>	Cria ficha de anamnese via link público com validação do token	Valida token, expiração, limite de usos e processa 30+ campos
<code>cancelar_eventos_serie(p_serie_id, p_a_partir_de)</code>	Cancela todos os eventos futuros de uma série recorrente	Não cancela eventos já realizados
<code>cancel_recurrence_from(p_recurrence_id, p_from_date)</code>	Encerra uma recorrência a partir de uma data, ajustando end_date	Se from_date ≤ start_date, cancela a recorrência inteira
<code>create_therapist_payout(...)</code>	Calcula e registra repasse financeiro para terapeuta em clínica	Agrega sessões realizadas no período com cálculo de comissão

Função	O que faz	Observação crítica
<code>create_support_session(p_tenant_id, p_ttl_minutes)</code>	Cria sessão temporária de acesso ao suporte (saas_admin only)	TTL: 1–120 minutos. Valida role = saas_admin
<code>can_delete_patient(p_patient_id)</code>	Verifica se é seguro deletar um paciente (sem eventos, recorrências, contratos)	Retorna boolean — chame antes de qualquer DELETE em patients

## Triggers Automáticos

- Sessão realizada → cobrança criada: Trigger `auto_create_financial_record_from_session` dispara quando `agenda_eventos.status` muda para "realizado". Cria `financial_record` com status "pending" e vencimento 7 dias após a sessão.
- Sessão cancelada → notificações canceladas: Trigger `cancel_notifications_on_session_cancel` cancela todas as notificações pendentes relacionadas à sessão.
- Opt-out do paciente → notificações canceladas: Trigger `cancel_notifications_on_opt_out` cancela notificações do canal específico quando paciente faz opt-out.

14

## Estratégia de Fases de Desenvolvimento

O desenvolvimento é organizado em quatro fases estratégicas. A infraestrutura de banco de dados (tenants, profiles, agenda, financeiro, notificações) já está substancialmente implementada — o foco atual é na experiência de produto sobre ela.

### FASE 1 MVP e Validação

Meses 1–3

- Sistema multi-papel funcional — profiles, tenants, tenant\_members implementados
- Agendamento completo com recorrência e lembretes via e-mail e WhatsApp
- Cadastro de pacientes com vínculo terapeuta-paciente e ficha de anamnese online
- Prontuário com anotações de sessão em agenda\_eventos.extra\_fields
- Financeiro básico: cobrança automática por trigger, integração Asaas (Pix)
- LGPD: criptografia, consentimento, auditoria via support\_sessions
- Beta fechado com 10–20 terapeutas reais
- Landing page de microlearning com 1 trilha inicial

### FASE 2 Diferenciação Clínica

Meses 4–8

- Escalas psicológicas integradas (PHQ-9, GAD-7, BDI) usando determined\_commitments com campos dinâmicos
- Dashboard de evolução do paciente com gráficos longitudinais
- Módulo do paciente: diário de humor, tarefas, visualização de agenda (PWA)
- Relatórios e laudos com template e exportação em PDF
- Módulo de Clínica: tenant kind clinic\_\*, repasse financeiro (therapist\_payouts)
- Expansão do microlearning: 3–5 trilhas completas
- Meta: 100 terapeutas pagantes, churn abaixo de 5%/mês

### FASE 3 Rede e Crescimento

Meses 9–18

- Diretório de terapeutas usando profiles públicos
- Rede de encaminhamentos — status "Encaminhado" em patients com rastreio
- Supervisão clínica — tenant kind supervisor + plano target=supervisor + max\_supervisees
- Programa de indicação com benefícios para terapeutas indicadores
- Expansão para clínicas multiprofissionais — kind clinic\_full
- Meta: 1.000 terapeutas ativos, 20+ clínicas, efeito de rede ativado

#### FASE 4 Inteligência e Escala

18+ meses

- IA de suporte clínico ingerindo agenda\_eventos.extra\_fields e histórico longitudinal
- Alertas de risco baseados em padrões comportamentais do paciente
- Integração com planos de saúde via insurance\_plans e insurance\_plan\_services (já no schema)
- API aberta para parceiros — profiles.platform\_roles já suporta roles de API
- Expansão internacional para mercados lusófonos (language e timezone já no schema)



15

# Cronograma do MVP (3 Meses)

Cronograma desenhado para um time com frontend e backend separados, priorizando entregas que gerem valor real para o primeiro usuário. A infraestrutura de banco já existe — o trabalho é de produto, UX e integração.

## MÊS 1 — Fundação Sólida

Período	Frontend	Backend	Entrega
Sem 1–2	Telas de login, cadastro, seleção de account_type, navegação base por papel	Auth Supabase, profiles criado no signup, middleware de permissão por role e account_type	Usuário cria conta, define se é terapeuta ou paciente e navega no ambiente correto
Sem 3–4	Dashboard por papel, perfil do usuário, criação de tenant (clínica/solo)	RPC create_clinic_tenant, tenant_members, RLS policies por tenant_id	Terapeuta cria workspace, convida colegas — multi-tenant funcionando

## MÊS 2 — Core do Produto

Período	Frontend	Backend	Entrega
Sem 1–2	Tela de agenda do terapeuta, configuração de disponibilidade (agenda_regras_semanais)	CRUD de agenda_regras_semanais, recurrence_rules, geração de agenda_eventos	Terapeuta configura agenda semanal com recorrência
Sem 3–4	Cadastro de pacientes, ficha de anamnese, link de auto-cadastro	CRUD patients, patient_invites, create_patient_intake_request_v2, notificações de agendamento	Paciente agenda sessão via link, terapeuta recebe notificação

## MÊS 3 — Fechar o Ciclo e Lançar

Período	Frontend	Backend	Entrega
Sem 1–2	Tela financeira, histórico de sessões, status de pagamento (financial_records)	Integração Asaas (Pix), trigger a uto_create_financial_record_fro m_session, webhooks	Ciclo financeiro completo: sessão realizada → cobrança automática → confirmação Pix
Sem 3–4	Anotações de sessão em extra_fields, UX refinado, landing page de microlearning	Revisão de segurança, RLS audit via support_sessions, ambiente de produção, monitoramento	Beta com 10–20 terapeutas reais

16

## Riscos e Mitigações

RLS mal configurada vazando dados entre tenants	<div>Impacto Alto</div> <div>Prob. Médio</div> <div>crítico de support_sessions p nova.</div>
Trigger financeiro falhando silenciosamente	<div>Impacto Médio</div> <div>Prob. Médio</div> <div>cre RA do Supabase. Impl como alerta.</div>
Integração de pagamento subestimada	<div>Impacto Médio</div> <div>Prob. Alto</div> <div>mer ebh payment_settings j</div>
Dados sensíveis sem criptografia	<div>Impacto Alto</div> <div>Prob. Baixo</div> <div>n e e Vault para secrets.</div>
Complexidade do multi-tenant na UI	<div>Impacto Médio</div> <div>Prob. Alto</div> <div>orta ra do onboarding para</div>
Adoção lenta pelos terapeutas	<div>Impacto Alto</div> <div>Prob. Médio</div> <div>om gra novidades in-app.</div>
Escopo crescente comprometendo o prazo	<div>Impacto Alto</div> <div>Prob. Alto</div> <div>ern tesis tenant_modules pa</div>

# Estratégia de Adoção e Crescimento

## O Funil de Aquisição

Topo do Funil — Conteúdo	Microlearning Presença no
Meio do Funil — Experimentação	Plano gratuito experimentar
Fundo do Funil — Conversão	Quando ultrapassar o checkout
Retenção — Profundidade	Escalas de cliques custo de suporte
Expansão — Indicação	Programa de modelo pago

## Estratégia de Beta — Os Primeiros 20 Terapeutas

- Recrutar manualmente via LinkedIn, grupos de psicólogos no WhatsApp e Facebook, indicações pessoais.
- Oferecer 6 meses gratuitos no plano Profissional em troca de uso semanal e feedback honesto — configurado via subscriptions com source = manual.
- Realizar call de 30 minutos com cada beta-tester após 2 semanas de uso.
- Observar onde travam no onboarding, o que tentam fazer e não encontram, o que elogiam espontaneamente.
- Usar esses aprendizados para uma rodada de ajustes de 2 semanas antes do lançamento público.
- Transformar os melhores beta-testers em embaixadores com benefícios permanentes.

18

## Visão de Longo Prazo

Em 5 anos, a plataforma deve ser reconhecida como a infraestrutura da saúde mental brasileira — o ambiente onde profissionais aprendem, trabalham, colaboram e crescem.

*Visão de longo prazo: Ser para o terapeuta brasileiro o que a Bloomberg é para o mercado financeiro — a plataforma que nenhum profissional sério consegue imaginar trabalhar sem ela.*

Plataforma de Referência Nacional	Educação Continuada Reconhecida	Certificações reconhecidas. Parcerias com conselhos e universidades. Base: platform_roles.
IA Clínica de Suporte	Expansão Lusófona	Portugal, Angola e Moçambique. O schema já contempla os fuso horários. profiles.timezone para o perfil.
Impacto Social Mensurável	Contribuir para o aumento da qualidade e acessibilidade do cuidado em saúde mental no Brasil, com métricas públicas de impacto.	

*Este documento é vivo. Deve ser revisado e atualizado a cada fase concluída, incorporando os aprendizados do mercado, as decisões do time e as evoluções do schema de banco de dados.*