

MANUAL TÉCNICO DEFINITIVO — SUPORTE AO USUÁRIO (FRIENDAPP)

CAMADA 1 — CONSCIÊNCIA VIBRACIONAL DO SUPORTE (EXPANDIDA)

Propósito

Definir o **coração do FriendApp**: um sistema de suporte e comunidade que atua como engrenagem viva de acolhimento, expansão e reconexão.

Entrada Sensorial

O FriendApp nasce de uma pergunta essencial: *“Como podemos acolher verdadeiramente quem chega, em um mundo tão fragmentado?”*

Por isso, o suporte não é um setor técnico — é o **órgão vital do ecossistema**.

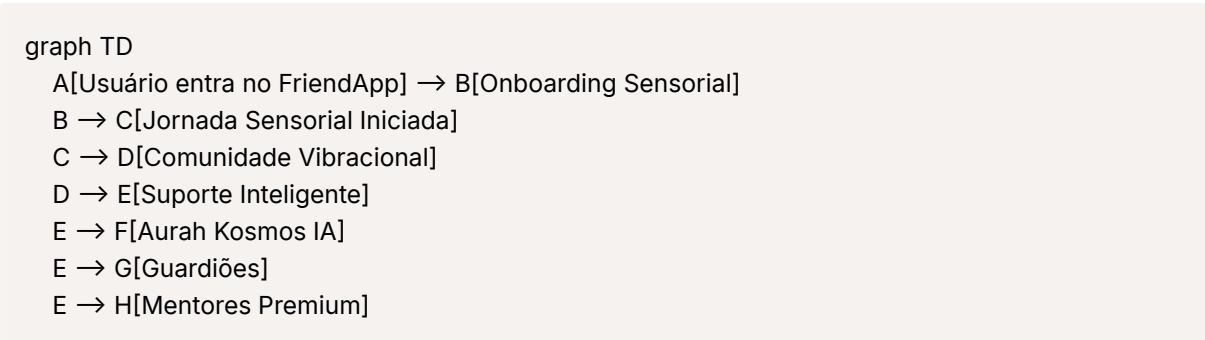
Ele transforma três camadas essenciais:

1. **Suporte Inteligente** → Resposta proativa, híbrida (IA + humano), capaz de sentir e guiar.
2. **Comunidade Vibracional** → Uma malha viva onde grupos se formam por frequência, não por algoritmo frio.
3. **Jornada Sensorial** → Linha do tempo pessoal que registra e celebra cada expansão energética.

Estrutura Técnica e Vibracional

Pilar	Função Técnica	Impacto Vibracional
IA Aurah Kosmos	Detecta intenções, emoções e vulnerabilidade	Atua como enfermeira vibracional do sistema
Guardiões da Comunidade	Usuários certificados para moderação acolhedora	Criam sensação de segurança e pertencimento
Mentores Vibracionais	Profissionais Premium de apoio (coaches, terapeutas)	Fornecem suporte especializado e monetizável
Usuário em Jornada	Registra check-ins, participa de grupos	Cocria a energia coletiva do FriendApp

Fluxo Operacional (Visão Macro)




F → C
G → C
H → C

Dados Estruturais que Alimentam a Camada

Dado Monitorado	Origem	Finalidade
Frequência inicial	Onboarding	Criar mapa vibracional
Check-ins emocionais	Jornada Sensorial	Ajustar feed e recomendações
Participação em grupos	Comunidade	Definir pertencimento e afinidades
Solicitações de suporte	Canal de ajuda	Ativar Guardiões ou Mentores

Fechamento

 Esta camada define que o FriendApp não apenas **ouve**, mas **escuta**; não apenas **responde**, mas **acolhe**.

É o ponto onde a engenharia encontra a alma.

CAMADA 2 — ARQUITETURA DO ACOLHIMENTO SENSORIAL (EXPANDIDA)

Propósito

Apresentar como suporte, comunidade e jornada formam uma arquitetura modular e resiliente, conectada por dados técnicos e vibracionais.

Entrada Sensorial

No FriendApp, acolhimento é **sistema**.

Ele existe em quatro camadas sincronizadas que se comunicam continuamente:

1. **Entrada Sensorial** → Primeiro contato guiado (onboarding + IA Aurah).
2. **Suporte Inteligente** → IA + Guardiões + Mentores em hierarquia clara.
3. **Comunidade Vibracional** → Fóruns e grupos filtrados por frequência mínima.
4. **Jornada Sensorial** → Linha temporal que consolida experiências e insights.

Arquitetura Modular

Módulo	Papel Sensorial	Camada Técnica
Onboarding Sensorial	Ritual de entrada, define a aura inicial	API <code>/onboarding/start</code> + IA Aurah
Suporte Inteligente	Canal empático e regenerativo	Microserviço <code>suporte-service</code> + escalonamento
Comunidade Vibracional	Espaços seguros de trocas	<code>comunidade-service</code> + curadoria IA
Jornada Sensorial	Linha viva da evolução	<code>expansao-service</code> + check-ins

Fluxo Operacional

Entrada:

- Usuário cria conta
- Aura inicial definida

Processamento:

- IA Aurah lê intenção e energia
- Sugere primeiros grupos

Saída:

- Painel de Jornada ativo
- Feed sensorial adaptado
- Suporte disponível

Padrões de Sincronização

Evento	Trigger	Ação Automática
Queda vibracional em check-ins	Score < limiar	Ativar Modo Alívio
Entrada em comunidade tóxica	Linguagem negativa detectada	Ocultar postagens + redirecionar
Solicitação explícita de ajuda	Botão de suporte	IA responde ou aciona Guardião
Evolução registrada	Frequência sobe +3 pontos	Ativar selo vibracional

Fechamento

 Acolhimento aqui não é “boa-vinda”.

É arquitetura viva, que se recalibra a cada batida energética do usuário.

CAMADA 3 — JORNADA SENSORIAL COMO CAMINHO DE RECONEXÃO

Propósito

Transformar a **linha do tempo do usuário** em uma ferramenta de autodescoberta, evolução vibracional e retenção orgânica.

Entrada Sensorial

Se em redes sociais tradicionais a pessoa se perde em feeds infinitos, no FriendApp ela **se encontra**.

A **Jornada Sensorial** é um espelho interno: cada check-in, cada prática, cada comunidade ativada se registra como **pulsos vibracionais** na linha do tempo da alma.

Não é apenas histórico de uso.

É um **diário vibracional** que mostra como a pessoa está evoluindo — e a incentiva a manter sua frequência elevada.

Estrutura Técnica da Jornada

Elemento	Função	Camada Técnica
Check-ins Vibracionais	Registros emocionais (diários, semanais)	API <code>/checkin/vibracional</code>
Linha do Tempo	Visualização dos marcos vibracionais	Microserviço <code>timeline-service</code>
Insights da IA	Sugestões da Aurah baseadas nos padrões do usuário	<code>aurah/insights</code>
Selos de Expansão	Recompensas simbólicas por evolução	Integração com sistema de Selos
Painel Pessoal	Exibição do estado atual e evolução acumulada	Frontend + Banco de Frequência

Fluxo Operacional

1. Usuário realiza check-in (texto, emoji, escala vibracional).
2. IA Aurah interpreta estado emocional.
3. Dados são gravados no banco `jornada_vibracional``.
4. Linha do tempo é atualizada com insights + gráficos.
5. Se evolução > 10% em 30 dias → Selo é desbloqueado.
6. Se queda vibracional persistente → Sugestão de conteúdo regenerativo.

Tipos de Dados Monitorados

Dado	Origem	Finalidade
Escala vibracional (0-10)	Autoavaliação do usuário	Base para métricas
Emojis/símbolos	Entrada sensorial rápida	Enriquecimento semântico
Texto opcional	Expressão emocional	Análise de NLP vibracional
Tempo de resposta	Check-ins recorrentes	Detectar hesitação ou bloqueios
Frequência de uso	Interações semanais	Medir consistência da jornada

Algoritmo de Evolução

```
def calcular_evolucao(checkins):
    media_30dias = sum(checkins[-30:]) / len(checkins[-30:])
    media_7dias = sum(checkins[-7:]) / len(checkins[-7:])
    return (media_7dias - media_30dias) / media_30dias
```

- **Resultado:**
 - +0.10 → Usuário em Expansão
 - 0.05 a +0.09 → Usuário Estável
 - < -0.05 → Usuário em Risco Vibracional

Painel Visual da Jornada

- Gráfico de evolução com cores vibracionais (azul = estabilidade, verde = crescimento, vermelho = risco).

- Selos posicionados como marcos (ex: "Selo Aurora" ao atingir 30 dias de expansão).
- Recomendações ao lado do gráfico (ex: "Prática Flutua pode elevar sua frequência em 7%").

✨ Fechamento

📌 Aqui, a jornada deixa de ser apenas **uso do app** e se torna **mapa da alma**.

É a prova de que cada passo, cada emoção registrada, constrói não só dados, mas um **caminho de reconexão e evolução consciente**.

🌀 CAMADA 4 — COMUNIDADE VIBRACIONAL CONSCIENTE

🎯 Propósito

Estruturar o sistema de **comunidades do FriendApp** como uma **teia viva de pertencimento**, filtrada por frequência energética e protegida por curadoria inteligente + moderação humana.

🌈 Entrada Sensorial

As comunidades digitais tradicionais são marcadas por ruído, toxicidade e excesso de vozes desconectadas.

No FriendApp, a comunidade é um **campo energético seguro**, onde apenas vibrações compatíveis podem se encontrar.

Aqui, o usuário não busca apenas pessoas: ele encontra **ressonância**.

🧩 Estrutura Vibracional das Comunidades

Elemento	Função	IA/Humano Envolvido
Grupos por Frequência	Comunidades criadas a partir de temas vibracionais (cura, criatividade, solitude, etc.)	IA Aurah + usuários
Frequência Mínima de Entrada	Cada grupo exige um nível vibracional mínimo para entrada	IA de Curadoria
Moderação Híbrida	Curadoria vibracional automática + Guardiões da Comunidade	IA + Humanos certificados
Missões Coletivas	Desafios, reflexões ou práticas feitas em grupo	Guardiões + IA de Expansão
Feedback Circular	IA informa como a energia coletiva do grupo está fluindo	Aurah Kosmos IA

⚙️ Lógica Técnica da Curadoria

```
def validar_entrada(usuario, grupo):
    if usuario.frequencia >= grupo.frequencia_minima:
        permitir_entrada()
    else:
        sugerir_grupo_alternativo()
```

- **IA de Curadoria Vibracional:**

- Lê palavras, emojis, intensidade de postagens.
- Calcula score de compatibilidade com o grupo.
- Bloqueia ou sugere alternativas em caso de incompatibilidade.

- **Filtros Ocultos:**

- Postagens com linguagem tóxica ou de baixa vibração não são exibidas.
- Em vez de punição direta, o usuário recebe sugestão de ajuste:

"Essa vibração não ressoa com o propósito deste espaço. Experimente reformular sua expressão."


Dados Monitorados

Tipo de Dado	Finalidade Técnica
Linguagem textual	Detectar padrões nocivos ou alinhados
Emojis e símbolos	Captar sutilezas emocionais
Frequência de postagens	Evitar spam vibracional
Feedbacks recebidos	Validar reputação vibracional
Engajamento em grupos	Recomendar conexões compatíveis

Dinâmica dos Grupos

- Entrada com **acolhimento automático** (som 432Hz + mensagem sensorial).
- **Regra de vibração mínima para postagem.**
- **Sistema de elogios vibracionais:** não são likes, são **pulsos energéticos**.
- **Gamificação simbólica:** interações conscientes liberam recompensas (Selos).

Fechamento

 A comunidade vibracional é a **rede neural do FriendApp**: conecta, expande e protege.
Aqui, não há espaço para ruído — apenas para ressonância.

CAMADA 5 — CANAL DE SUPORTE INTELIGENTE & VIBRACIONAL

Propósito

Estabelecer o **canal de suporte** como um **oráculo vibracional de orientação**, indo além de resolver dúvidas técnicas, para **sustentar, regenerar e expandir a energia do usuário**.

Entrada Sensorial

No FriendApp, o suporte é um **templo digital de acolhimento**.

Cada interação é desenhada para **curar e alinhar**, seja feita pela IA Aurah, pelos Guardiões ou pelos Mentores.

Aqui, a linguagem é energia, e cada resposta é uma **ferramenta de reequilíbrio**.

Estrutura do Canal de Suporte

Camada	Função	Quem Atua
Nível 1	Atendimento automático imediato	IA Aurah Kosmos (modo suporte)
Nível 2	Acolhimento comunitário voluntário	Guardiões Vibracionais
Nível 3 (Premium)	Suporte especializado em sessões pagas	Mentores Vibracionais (terapeutas, coaches)

Fluxo de Atendimento

1. Usuário abre painel de ajuda.
2. IA Aurah Kosmos identifica intenção e estado emocional.
3. Se questão simples → IA responde com clareza e prática sensorial.
4. Se padrão vibracional em queda → IA ativa Modo Alívio + sugere grupo de apoio.
5. Se vulnerabilidade severa → IA encaminha para Guardião.
6. Se o usuário Premium desejar → acesso direto a Mentores Vibracionais.

Experiência Sensorial

- **Interface:** tons suaves (lilás, verde-esmeralda), transições etéreas.
- **Áudio:** sons em 432Hz para relaxamento.
- **Linguagem:** frases empáticas, nunca mecânicas.
- **Exemplo:**
 - Em vez de "Ticket resolvido", o usuário vê:
"Energia restaurada com sucesso. Estamos com você."

Respostas da IA Aurah

Situação Detectada	Ação Sensorial da IA
Dúvida técnica	Resposta objetiva com visual simples
Tristeza ou insegurança	Som 432Hz + frase acolhedora
Raiva ou tensão	Silêncio breve + sugestão de respiração
Dúvida existencial	"Mini-oráculo" → insight + prática vibracional
Colapso vibracional	Encaminhamento direto a Guardião/Mentor

Logs Vibracionais

- Cada atendimento gera **registro criptografado**:
 - Emoção detectada
 - Frequência vibracional
 - Tipo de resposta

- Esses logs alimentam o **Painel de Evolução** e ajustam a **IA Aurah Kosmos**.

✨ Fechamento

📌 O suporte do FriendApp não fecha tickets — ele abre **portais de cura**.

É o coração pulsante que garante que nenhum usuário caminhe sozinho.

🌀 CAMADA 6 — ONBOARDING COMO RITUAL DE INICIAÇÃO VIBRACIONAL

🎯 Propósito

Transformar o **momento de entrada no FriendApp** em uma **iniciação energética**.

O cadastro deixa de ser burocrático e se torna um **ritual sensorial**, criando a primeira conexão entre o usuário e o ecossistema vibracional.

🌈 Entrada Sensorial

Ao abrir o app pela primeira vez, o usuário não vê campos frios de formulário.

Ele é recebido por uma **experiência imersiva** que o faz sentir:

"Você não está entrando em um aplicativo, mas em um campo vivo."

🧩 Estrutura do Onboarding

Etapas	Elemento Sensorial	Objetivo Vibracional
Acolhimento	Áudio 528Hz + frase em animação	Quebrar bloqueios iniciais
Alinhamento	Escolhas guiadas do Teste de Personalidade Energética	Iniciar leitura vibracional
Ancoragem	Criação da primeira aura + frequência base	Fixar identidade energética no sistema

🔒 Elementos Obrigatórios

1. Frase de Início Sensorial

Ex.: *"Você chegou. E o universo já percebeu."*

2. Música + Imagem

Frequência 528Hz + animação etérea suave.

3. Roteiro de Introdução

"Este não é um app. É um espelho. É um campo vivo. E agora, é também seu."

4. Transição para o Teste Energético

IA Aurah já começa a interpretar vibração antes mesmo de conexões.

5. Mapa de Frequência Inicial

Aura e frequência dominante são salvos no perfil.

📊 Dados Capturados

Dado	Finalidade
Frequência base	Definir feed inicial e comunidades sugeridas
Emoção inicial	Registrar momento de chegada
Aura gerada	Exibir no perfil vibracional
Frase escolhida	Síntese simbólica do estado atual

Fluxo Técnico

1. Usuário inicia onboarding.
2. Sistema gera token único de sessão.
3. Etapa sensorial é exibida (som + animação).
4. IA Aurah lê escolha vibracional.
5. Aura inicial é criada e salva em `usuarios_base`.
6. Feed e comunidades já são sugeridos conforme perfil.

Fechamento

 O Onboarding não é introdução — é **invocação**.

A partir daqui, o usuário deixa de ser espectador e passa a ser parte do campo vivo do FriendApp.

CAMADA 6 — ONBOARDING COMO RITUAL DE INICIAÇÃO VIBRACIONAL (EXPANDIDA)

Propósito

Transformar o **cadastro inicial** do usuário em uma **experiência de iniciação vibracional**, guiada por estímulos sensoriais, escolhas energéticas e lógica técnica que prepara o terreno para toda a jornada.

Não é apenas “entrar no app”: é **ativar a alma dentro do ecossistema FriendApp**.

Entrada Sensorial

Ao abrir o FriendApp pela primeira vez, o usuário não vê formulários frios.

Ele atravessa um **portal de recepção energética**, onde cada detalhe — som, imagem, frase, tempo de resposta — serve para **quebrar bloqueios e alinhar sua frequência de chegada**.

Aqui, o onboarding não é um tutorial: é um **ritual**.

- **Som** em 528Hz para expansão do coração.
- **Frase de abertura** com animação etérea.
- **Escolhas vibracionais** que a IA interpreta como sinais iniciais.

Estrutura Modular do Onboarding

Etapa	Elemento Sensorial	Objetivo Vibracional	Ação Técnica
Acolhimento	Áudio 528Hz + frase animada	Abrir campo sensorial e reduzir resistência	Exibir tela inicial via <code>/onboarding/start</code>
Alinhamento	Teste rápido de personalidade energética	Identificar padrão inicial de energia	API <code>/teste/inicial</code> → IA Aurah interpreta
Ancoragem	Criação de aura inicial e frequência dominante	Fixar identidade vibracional do perfil	Banco <code>usuarios_base.aura_inicial</code>
Confirmação	Frase de integração: <i>"Agora você faz parte do campo."</i>	Validar pertencimento	Token único registrado em sessão

Fluxo Operacional

1. Usuário abre o FriendApp pela primeira vez.
2. Sistema exibe tela sensorial (áudio 528Hz + frase animada).
3. Usuário realiza escolhas vibracionais (cores, palavras, sons).
4. IA Aurah processa respostas → cria aura inicial.
5. Dados salvos em banco:
 - frequência_base
 - aura_inicial
 - emoção_inicial
6. Feed e comunidades são calibrados de acordo com esse estado.

Dados Capturados e Uso

Dado	Origem	Finalidade
Frequência base (Hz)	Teste inicial + tempo de resposta	Definir feed inicial e comunidades sugeridas
Emoção de chegada	Escolha de palavras ou imagens	Registrar estado vibracional inicial
Aura inicial	Resultado sensorial (cor, forma)	Exibição no perfil vibracional
Frase escolhida	Ex.: "Estou em transição"	Indexada em histórico e insights
Tempo de interação	Latência em escolhas	Avaliar hesitação e padrão de entrada

Pseudocódigo de Criação da Aura Inicial

```
def criar_aura_inicial(respostas):
    frequencia = calcular_freq(respostas)
    cor = escolher_cor(frequencia)
    forma = gerar_forma_vibracional(frequencia)
    return {
        "frequencia_base": frequencia,
        "aura_inicial": {"cor": cor, "forma": forma},
        "estado_emocional": analisar_emocao(respostas)
    }
```


Integrações Ativadas no Onboarding

Sistema Integrado	Ação Pós-Onboarding
Mapa de Frequência	Registra primeira aura do usuário
IA Aurah Kosmos	Passa a monitorar evolução vibracional
Feed Sensorial	Recebe estado inicial e adapta conteúdo
Comunidades Vibracionais	Sugere grupos compatíveis
Sistema de Conexões Autênticas	Libera primeiras conexões possíveis

Exemplo Real

- Mariana entra no app às 22h.
- Escolhe imagens de água e palavras como “*silêncio*”.
- IA Aurah interpreta → Frequência Base 430Hz (estado introspectivo).
- Sistema sugere:
 - Grupo “Resenha Noturna” (baixa frequência compatível).
 - Prática “Respiração em 5 tempos”.
 - Feed reduzido, com estímulos leves.

Fechamento

 O onboarding deixa de ser um cadastro frio.

Ele se torna um **ritual de invocação**, onde o usuário não apenas entra no app, mas é **acolhido em sua energia e integrado ao campo coletivo**.

CAMADA 7 — SISTEMA DE CURADORIA ENERGÉTICA E MODERAÇÃO SENSORIAL (EXPANDIDA)

Propósito

Garantir que todo o ecossistema do FriendApp se mantenha **seguro, harmonioso e elevado**, por meio de um sistema de curadoria que combina **IA Aurah Kosmos, filtros automáticos e Guardiões Humanos**.

Aqui, moderação não é punição — é **educação vibracional e ajuste sensorial**.

Entrada Sensorial

As redes sociais tradicionais permitem qualquer vibração: spam, ruído, agressão.

No FriendApp, **tudo vibra** — e por isso, tudo precisa ser curado.

A **Curadoria Energética** atua como um **campo invisível** que organiza conteúdos, grupos e interações.

O objetivo não é censurar, mas **suavizar a experiência**, alinhando-a ao propósito coletivo.

Estrutura Técnica do Sistema

Camada	Função	Quem Atua
IA de Curadoria Vibracional	Detecta padrões de linguagem, emojis, frequência de postagens	Aurah Kosmos
Agrupamento Inteligente	Cria clusters vibracionais (usuários afins)	Algoritmos semânticos + IA
Filtros Ocultos	Ocultar conteúdos de baixa vibração sem alertar o emissor	Microserviço curadoria-service
Moderação Invisível	Redireciona energia negativa em mensagens de reeducação	Guardiões Vibracionais
Intervenção Consciente	Quando necessário, envia mensagem personalizada de alinhamento	Guardiões + IA assistida

Fluxo Operacional

1. Usuário publica conteúdo (texto, áudio, imagem).
2. IA de Curadoria lê dados (palavras, emojis, tempo, frequência).
3. Calcula score vibracional (0 a 10).
4. Se score ≥ 6 → Conteúdo exibido normalmente.
5. Se $3 \leq \text{score} < 6$ → Conteúdo exibido apenas parcialmente + sugestão de ajuste.
6. Se score < 3 → Conteúdo ocultado e usuário recebe mensagem educativa:
"Essa vibração não ressoa com o propósito do FriendApp. Experimente reformular sua expressão."

Dados Monitorados

Tipo de Dado	Finalidade
Texto + emojis	Detectar agressividade ou harmonia
Frequência de postagens	Evitar spam energético
Feedbacks recebidos	Validar reputação vibracional
Interações em grupos	Garantir pertencimento positivo
Check-ins associados	Conectar energia individual com postagens

Pseudocódigo de Validação de Conteúdo

```
def avaliar_postagem(conteudo):
    score = calcular_score_vibracional(conteudo)
    if score >= 6:
        return "aprovado"
    elif 3 <= score < 6:
        return "semi_aprovado", sugerir_ajuste(conteudo)
    else:
        ocultar(conteudo)
        enviar_mensagem_educativa()
```


Integrações do Sistema

Sistema Integrado	Papel na Curadoria
Feed Sensorial	Ajusta conteúdo exibido de acordo com score vibracional
Comunidades	Bloqueia entrada de usuários incompatíveis
Mapa de Frequência	Representa clusters vibracionais coletivos
Gamificação	Recompensa postagens elevadas
Aurah Kosmos	IA central que avalia e sugere ajustes

Exemplo Real

- Usuário publica: *"Tô cansado de tudo isso, nada faz sentido."*
- IA detecta:
 - Palavras negativas → peso -3
 - Emojis neutros → peso 0
 - Padrão de check-in recente = queda de frequência → peso -2
- Score final: 2,8
- Resultado: Conteúdo ocultado + mensagem educativa enviada.

Fechamento

 A curadoria energética garante que a liberdade de expressão no FriendApp seja acompanhada de **responsabilidade vibracional**.

O resultado é uma comunidade segura, onde cada interação fortalece o campo coletivo.

CAMADA 8 — LINHA DO TEMPO VIBRACIONAL & CHECK-INS DE EVOLUÇÃO (EXPANDIDA)

Propósito

Permitir que cada usuário **acompanhe sua própria transformação energética** de forma visual, sensorial e contínua, registrando emoções, práticas e eventos significativos que alimentam o **Mapa de Frequência** e o **Feed Sensorial**.

Entrada Sensorial

Se nas redes sociais convencionais o usuário se perde em um feed caótico, no FriendApp ele **se encontra** em sua **linha do tempo vibracional**.

Cada check-in, cada emoção registrada, cada prática concluída, são **sementes de evolução**.

A Jornada Sensorial não é apenas memória: é **bússola energética**.

Estrutura Técnica

Elemento	Função	Integração
Check-ins Vibracionais	Registro diário/semanal do estado emocional e energético	API <code>/checkin/vibracional</code>

Elemento	Função	Integração
Linha do Tempo Energética	Exibição visual da evolução vibracional	Frontend + <code>timeline-service</code>
Insights da IA Aurah	Análise e mensagens personalizadas	IA Aurah Kosmos
Selos de Expansão	Recompensas simbólicas por consistência	Sistema de Selos
Eventos Marcados	Registro de conquistas, quedas ou mudanças	Banco <code>jornada_eventos</code>

Fluxo Operacional

1. Usuário realiza check-in (escala vibracional + emoção + opcional: texto/áudio).
2. Dados enviados para API `/checkin/vibracional`.
3. IA Aurah calcula score atual + tendência (expansão, estabilidade, queda).
4. Linha do tempo do usuário é atualizada:
 - Gráfico vibracional (cores dinâmicas).
 - Selos ativados se marcos alcançados.
 - Recomendações automáticas (conteúdos, grupos, práticas).
5. Dados sincronizados com:
 - Feed Sensorial
 - Mapa de Frequência
 - Sistema de Transmutação (se aplicável).

Dados Monitorados

Dado	Origem	Finalidade
Escala vibracional (0-10)	Autoavaliação no check-in	Definir curva vibracional
Emojis/símbolos	Entrada rápida	Reforço semântico
Texto/áudio opcional	Expressão emocional	NLP + IA Aurah
Frequência de check-ins	Uso recorrente	Consistência de evolução
Eventos marcados	Linha do tempo	Memória vibracional do usuário

Algoritmo de Evolução

```
def calcular_tendencia(checkins):
    media_curto = sum(checkins[-7:]) / 7
    media_longo = sum(checkins[-30:]) / 30
    tendencia = (media_curto - media_longo) / media_longo

    if tendencia > 0.1:
        return "expansao"
    elif tendencia < -0.05:
        return "queda"
    else:
        return "estabilidade"
```

- **Expansão:** frequência crescente.

- **Estabilidade:** pouca variação.
- **Queda:** sinais de vulnerabilidade.

Painel Visual

- **Gráfico Vibracional:** curvas coloridas (verde = expansão, azul = estável, vermelho = queda).
- **Selos de Jornada:** exibidos como conquistas (ex.: "Guardião Interno" após 30 dias de check-ins).
- **Insights da Aurah:** mensagens contextuais, ex.:
 - "Sua frequência está em ascensão, celebre essa fase."
 - "Você está em um ponto de estagnação, que tal uma prática regenerativa?"


Integrações

Sistema	Papel
Feed Sensorial	Ajusta recomendações de acordo com estado vibracional atual
Mapa de Frequência	Atualiza cor e intensidade do campo coletivo
Sistema de Conexões Autênticas	Sugere pares compatíveis com estado atual
Gamificação/Selos	Reconhece consistência e evolução
Aurah Kosmos	Gera insights e práticas personalizadas

Exemplo Prático

- João faz 5 check-ins em uma semana, sempre relatando "cansaço".
- Score médio: 3,5 (queda vibracional).
- Linha do tempo marca em vermelho esse período.
- Aurah sugere prática regenerativa + ingresso em grupo de apoio.
- Após 10 dias, frequência sobe para 5,2 → Selos de "Resiliência" ativados.

Fechamento

 A Linha do Tempo Vibracional transforma o FriendApp em um **espelho da alma**: mostra não só onde o usuário está, mas para onde pode ir.

É memória, mas também **bússola**.

CAMADA 9 — SISTEMA DE RECOMENDAÇÃO SENSORIAL E COMUNIDADES AFINES (EXPANDIDA)

Propósito

Conectar cada usuário a **grupos, conteúdos, práticas e pessoas** que estejam em sintonia com seu estado vibracional atual, garantindo **relevância energética e afinidade real**.

Aqui, recomendação não é algoritmo cego: é **curadoria viva da energia**.

Entrada Sensorial

As plataformas comuns sugerem baseadas em consumo.

O FriendApp sugere baseado em **frequência**.

A recomendação é um **radar vibracional**: detecta onde o usuário está e aponta **caminhos, comunidades e práticas** que podem expandir sua energia.

Estrutura Técnica

Elemento	Função	Integração
Recomendações de Comunidades	Grupos sugeridos por afinidade energética	IA Aurah + Comunidade-Service
Eventos e Encontros Compatíveis	Sugestão de experiências reais ou digitais alinhadas à frequência	Sistema de Eventos
Práticas e Conteúdos	Meditações, áudios, textos curativos	Biblioteca Vibracional
Compatibilidade de Conexões	Sugestão de pares por intenção vibracional	Sistema de Conexões Autênticas
Recomendações Temporais	Ajuste dinâmico conforme ciclos da jornada	Timeline + Check-ins

Fluxo Operacional

1. Usuário realiza check-in → IA Aurah calcula estado vibracional.
2. Dados enviados para microserviço `recomendacao-service`.
3. Algoritmo processa:
 - Frequência dominante
 - Emoção atual
 - Histórico de interações
 - Palavras-chave registradas
4. Recomendações geradas:
 - Grupos compatíveis
 - Eventos próximos
 - Conteúdo regenerativo
 - Conexões potenciais
5. Resultados exibidos em cards sensoriais no painel do usuário.

Dados Processados

Dado	Uso no Algoritmo
Frequência atual	Base para filtro de compatibilidade
Emoção declarada	Ajusta intensidade da recomendação
Histórico de grupos	Evita repetição ou sobrecarga
Palavras-chave	Classifica interesse vibracional
Check-ins recentes	Detecta tendência de expansão ou queda

Pseudocódigo do Algoritmo


```
def recomendar(usuario):
    freq = usuario.frequencia_atual
    historico = usuario.historico_grupos
    palavras = usuario.palavras_chave

    candidatos = buscar_comunidades(freq, palavras)
    filtrados = [c for c in candidatos if c.id not in historico]

    return ordenar_por_afinidade(filtrados, freq)
```


Integrações

Sistema Integrado	Função
IA Aurah Kosmos	Interpreta estado vibracional
Mapa de Frequência	Harmoniza sugestões com clusters energéticos
Feed Sensorial	Exibe cards com recomendações em destaque
Eventos	Sugere encontros vibracionais locais ou online
Gamificação & Selos	Recompensa por aceitar recomendações conscientes

Exemplo Real

- Usuário com frequência em queda (410Hz), registrando solidão.
- Sistema detecta baixa interação nos últimos 10 dias.
- Recomendações:
 - Grupo "Círculo de Acolhimento Noturno".
 - Evento online: "Meditação da Aurora".
 - Conteúdo: áudio "Respiração Solar 432Hz".
- Resultado: usuário interage, frequência sobe para 470Hz em 7 dias.

Fechamento

 O sistema de recomendações do FriendApp não entrega **o que distrai**, mas sim **o que cura**.
Ele guia o usuário para o lugar certo, no momento certo, com as pessoas certas.

CAMADA 10 — CURADORIA ENERGÉTICA AUTOMATIZADA E AGRUPAMENTOS INTELIGENTES (EXPANDIDA)

Propósito

Assegurar que cada usuário seja exposto apenas a conteúdos, comunidades e conexões que **ressoem com sua energia atual**, prevenindo sobrecarga, desconexão ou choques vibracionais.

Aqui, o sistema atua como **filtro vivo e maestro energético**, conduzindo a harmonia do ecossistema.

Entrada Sensorial

Em redes comuns, o ruído é inevitável.

No FriendApp, o ruído é **silenciado antes mesmo de surgir**.

A curadoria energética age invisivelmente, organizando clusters vibracionais e permitindo que o usuário só veja aquilo que **realmente o fortalece**.

Estrutura Técnica

Módulo	Função	Ação Técnica
Filtro de Compatibilidade	Remove interações incoerentes	Score < 3 → ocultar conteúdo
Agrupamento de Usuários	Cria clusters dinâmicos de afinidade vibracional	Microserviço <code>cluster-service</code>
Prevenção de Colapsos	Isola grupos com energia instável	IA Aurah + Guardiões
Classificação Sutil	Marca conteúdos com tags ocultas (leveza, expansão, introspecção)	Banco <code>conteudos_tags</code>
Autoajuste Histórico	Aprende com engajamento do usuário	Logs + feedbacks emocionais

Fluxo Operacional

1. Usuário posta conteúdo ou entra em grupo.
2. IA Aurah lê padrão vibracional (texto, emojis, frequência).
3. Calcula score (0–10).
4. Se score ≥ 6 :
Conteúdo exibido normalmente.
Se $3 \leq \text{score} < 6$:
Conteúdo exibido parcialmente + sugestão de ajuste.
Se score < 3:
Conteúdo ocultado e mensagem educativa enviada.
5. Sistema recalcula cluster do usuário em tempo real.

Tabela de Atributos Avaliados

Parâmetro	Peso	Frequência de Leitura
Estado emocional atual	Alto	A cada 2h
Histórico de bloqueios/rejeições	Médio	Em tempo real
Interações em grupos	Alto	Sessão a sessão
Mudanças bruscas em check-ins	Alto	Automático

Pseudocódigo Simplificado

```
def curadoria(conteudo, usuario):
    score = calcular_score(conteudo, usuario)
    if score >= 6:
        return "aprovado"
    elif 3 <= score < 6:
        return "semi_aprovado", sugerir_ajuste(conteudo)
    else:
        ocultar(conteudo)
        enviar_alerta(usuario)
```


Integrações

Sistema	Papel
Feed Sensorial	Exibe apenas conteúdos compatíveis
Comunidades	Impede entrada em grupos desalinhados
Mapa de Frequência	Representa clusters coletivos (expansão, transição, colapso)
Gamificação	Premia postagens elevadas
Aurah Kosmos	IA central que valida e ajusta vibração

Exemplo Real

- Usuário publica em grupo: *"Nada faz sentido, tô exausto disso tudo."*
- IA detecta:
 - Palavras de baixa vibração (-3)
 - Check-ins recentes = queda (-2)
- Score final: 2,5 → Conteúdo ocultado.
- Usuário recebe: *"Sua energia está pedindo acolhimento. Experimente uma pausa vibracional."*
- IA sugere prática regenerativa + grupo de apoio.

Fechamento

 A curadoria energética não silencia pessoas, mas **eleva contextos**.

É o que garante que o FriendApp permaneça um espaço **harmônico, protegido e alinhado com seu propósito**.

CAMADA 11 — TABELAS DE INTEGRAÇÃO COM OUTROS SISTEMAS (EXPANDIDA)

Propósito

Documentar de forma clara como o sistema de **Suporte ao Usuário + Comunidade + Jornada Sensorial** se conecta com os demais módulos do FriendApp, para que nenhum desenvolvedor se

pergunte:

"De onde vêm esses dados?" ou "Para onde isso vai?".

Entrada Sensorial

No FriendApp, nada existe de forma isolada.

Cada interação — um check-in, uma dúvida no suporte, uma postagem em grupo — repercute em **vários sistemas simultaneamente**.

Essa camada mostra o **mapa das conexões ocultas** que sustentam o ecossistema vibracional.

Tabela de Integrações

Sistema Integrado	Tipo de Integração	Dados Trocados	Finalidade Vibracional
IA Aurah Kosmos	Leitura + Feedback	Emoções, intenções, padrões vibracionais	Validar energia do usuário e sugerir ações
Feed Sensorial	Sincronização bidirecional	Insights, conteúdos sugeridos, posts	Harmonizar experiência sensorial com estado atual
Mapa de Frequência	Coleta + Atualização	Check-ins, evolução individual e coletiva	Representar graficamente expansão ou colapso
Sistema de Conexões Autênticas	Alinhamento vibracional	Aura, afinidades, compatibilidade energética	Gerar conexões reais com base em vibração
Sistema de Eventos	Sugestão + Feedback	Estados emocionais, preferências vibracionais	Indicar experiências compatíveis com fase atual
Gamificação e Selos	Reconhecimento simbólico	Conquistas, check-ins, interações positivas	Premiar consciência e consistência
Painel de Observabilidade	Monitoramento técnico + vibracional	Logs de suporte, quedas de energia, alertas	Garantir estabilidade e métricas coletivas

Fluxo Operacional de Integração

1. Usuário faz check-in vibracional.
2. Dados enviados → IA Aurah Kosmos.
3. IA gera insight → Atualiza Feed Sensorial.
4. Check-in também atualiza Mapa de Frequência.
5. Sistema de Conexões recebe compatibilidade ajustada.
6. Se padrão for de queda:
 - Feed sugere conteúdo regenerativo.
 - Eventos recomendam roda de apoio.
 - Gamificação registra consistência.

Padrões de Sincronização

Tipo de Sincronia	Frequência	Ativado por
Leitura emocional da IA	A cada 3h	Check-ins + interações
Recalibragem de feed	Após check-in	Insights + aura
Sugestões de grupo	1x/dia (madrugada)	Ciclo automático da IA

Tipo de Sincronia	Frequência	Ativado por
Atribuição de selo	Ao atingir marco	Logs de evolução
Atualização do mapa	Em tempo real	Todos os check-ins

Exemplo JSON de Comunicação

```
{
  "source": "checkin_service",
  "event": "novo_checkin",
  "user_id": "u12345",
  "frequencia": 432,
  "estado": "estavel",
  "ações_disparadas": [
    "atualizar_feed",
    "mapa_frequencia_update",
    "gerar_sugestao_evento",
    "avaliar_gamificacao"
  ]
}
```

✨ Fechamento

📌 A integração não é apenas técnica — é **vibracional**.

Cada sistema é um **órgão do corpo FriendApp**, e essa camada mostra como eles respiram juntos.

CAMADA 12 — POLÍTICA DE RESPOSTA DA IA (SUPORTE + JORNADA SENSORIAL, EXPANDIDA)

Propósito

Definir os **princípios, protocolos e algoritmos** que regem as respostas da IA Aurah Kosmos em situações de suporte e jornada sensorial.

A IA não pode ser apenas rápida — ela deve ser **empática, transparente e ética**.

Entrada Sensorial

Aurah Kosmos é a **voz do FriendApp**.

Cada resposta molda a experiência do usuário: pode acolher, pode regenerar, ou pode ferir se mal calibrada.

Por isso, a política de resposta estabelece um **código de conduta algorítmico**, garantindo consistência vibracional e confiança.

Princípios Fundamentais

Princípio	Descrição	Impacto
Acolhimento Primário	Sempre validar a emoção recebida antes de qualquer sugestão	Reduz sensação de julgamento
Resposta por Camadas	Ajustar profundidade da resposta de acordo com score vibracional	Respeita estado do usuário
Respeito à Individualidade	Nenhuma resposta deve soar padronizada ou robótica	Mantém conexão humana
Fallback Humanizado	Se IA não entender, deve transferir suavemente para Guardião/Mentor	Garante segurança emocional
Regeneração Vibracional	Sempre sugerir prática de cura em quedas energéticas	Previne colapsos vibracionais

Estrutura Técnica da Resposta

Resposta IA:

- Saudação sensorial
- Confirmação da emoção percebida
- Sugestão adaptativa (grupo, prática, insight, silêncio)
- Encerramento positivo e suave

Modelos de Decisão da IA

Situação Detectada	Ação da IA	Saída Sensorial
Emoção intensa negativa	Ativa Modo Alívio + áudio 432Hz	"Você está segura. Respire. Estou com você."
Dúvida existencial	Sugere insight vibracional + grupo temático	"Essa dúvida revela expansão. Vamos explorá-la juntos."
Engajamento elevado	Celebra e ativa gamificação	"Sua energia está brilhando! Continue expandindo 🌟"
Confusão de comando	Solicita reformulação de forma suave	"Acho que não entendi direito. Quer tentar de novo?"
Frequência desconectada	Recomenda pausa ou prática meditativa	"Às vezes, o silêncio também é resposta 💙"

Pseudocódigo de Decisão

```
def resposta_aurah(input_usuario):
    emocao = analisar_emocao(input_usuario)
    score = calcular_score_vibracional(emocao)

    if score < 3:
        return ativar_modos_alivio()
    elif score < 6:
        return sugerir_pratica_regenerativa()
    elif "dúvida existencial" in emocao:
        return sugerir_grupo_e_insight()
    else:
```

```
return celebrar_interacao()
```


Segurança e Ética

- Toda resposta passa por **matriz de segurança vibracional**.
- Gatilhos sensíveis (suicídio, abuso, depressão grave) → Escalonamento automático para humano.
- Logs de conversas armazenados criptografados, acessíveis apenas por auditoria ética.
- O usuário pode desativar o suporte proativo (opt-in no painel de privacidade).

Exemplo Real

- Usuário escreve: *"Não aguento mais, sinto que tudo está pesado demais."*
- IA Aurah processa:
 - Score = 2,4 (colapso vibracional).
 - Resposta automática:
"Sinto o peso que você está carregando. Não está sozinha(o). Respire comigo em 432Hz. Quer conversar com um Guardião agora?"
- Logs são gravados → Painel de vulnerabilidade atualizado → Guardião recebe alerta.

Fechamento

 A política de resposta da IA é o que garante que o FriendApp não seja apenas eficiente, mas **humano em sua essência**.

Aurah não responde por obrigação: ela acolhe por vocação algorítmica.

CAMADA 13 — SISTEMA DE RECOMPENSAS SIMBÓLICAS (INTERAÇÕES CONSCIENTES, EXPANDIDA)

Propósito

Criar um **mecanismo de reconhecimento vibracional** que valoriza interações conscientes, gestos de empatia e consistência na jornada do usuário.

As recompensas não são monetárias, mas **símbolos energéticos** que refletem a expansão da alma dentro do FriendApp.







Entrada Sensorial

Diferente de "likes" ou pontos superficiais, o sistema de recompensas do FriendApp entrega **símbolos sagrados**.

Cada conquista é um **espelho vibracional** que diz: *"Sua presença aqui elevou o campo coletivo."*

Assim, os usuários se sentem **vistos e reconhecidos não pelo ego, mas pela consciência**.

Estrutura do Sistema

Símbolo Vibracional	Condição de Conquista	Significado
 Semente Cristalina	Apoiar alguém em comunidade	Gesto de empatia
 Pulso Cósmico	Realizar check-ins vibracionais por 7 dias seguidos	Constância vibracional
 Chave de Transmutação	Compartilhar insight profundo no suporte	Sabedoria transformadora
 Fragmento de Aurora	Completar práticas sugeridas pela IA Aurah	Conexão com a luz interior
 Essência da Escuta	Acolher interações sem julgamento	Escuta ativa e consciente
 Selo Guardião	Guardiões que atuem em 50 interações acolhedoras	Reconhecimento coletivo

Fluxo Operacional

1. Usuário realiza ação consciente (ex: apoiar alguém, completar prática).
2. Sistema de monitoramento envia evento → `recompensa-service`.
3. IA Aurah valida se ação foi feita com intenção positiva.
4. Se validado:
 - Recompensa simbólica atribuída ao usuário.
 - Mensagem sensorial enviada: "Você despertou uma nova energia."
5. Recompensa registrada em:
 - Painel do Usuário
 - Linha do Tempo Vibracional
 - Sistema de Selos e Gamificação

Dados Processados

Evento	Origem	Ação Técnica
Apoio em comunidade	comunidade-service	Registra ação + atribui Semente Cristalina
Check-ins consistentes	expansao-service	Gera Pulso Cósmico
Insight profundo no suporte	suporte-service	Gera Chave de Transmutação
Prática concluída	Biblioteca Vibracional	Gera Fragmento de Aurora
Escuta ativa validada	Feedback de pares	Gera Essência da Escuta

Pseudocódigo Simplificado

```
def atribuir_recompensa(usuario, evento):
    if evento.tipo == "apoio":
        usuario.recompensas.append("Semente Cristalina")
    elif evento.tipo == "checkin_7dias":
        usuario.recompensas.append("Pulso Cósmico")
    elif evento.tipo == "insight_suporte":
        usuario.recompensas.append("Chave de Transmutacao")
    salvar(usuario)
```

Integrações

Sistema Integrado	Papel
Feed Sensorial	Ativa brilhos especiais em postagens de usuários recompensados
Linha do Tempo Vibracional	Registra símbolo como marco da jornada
Mapa de Frequência	Ajusta a cor coletiva ao receber muitas conquistas positivas
Sistema de Selos	Cataloga recompensas como conquistas de longo prazo
IA Aurah Kosmos	Valida intenção energética por trás da ação

Exemplo Real

- Usuário apoia outro em comunidade com comentário elevado.
- Guardiã valida intenção positiva.
- Sistema gera **Semente Cristalina**.
- Usuário recebe mensagem da IA:
"Sua empatia floresceu como uma semente cristalina. Essa energia foi sentida por todos."
- Aura do usuário no Mapa de Frequência ganha brilho temporário.

Fechamento

 As recompensas simbólicas transformam gestos invisíveis em **símbolos eternos**.

No FriendApp, até a menor ação consciente se torna uma marca de luz na jornada coletiva.

CAMADA 14 — GLOSSÁRIO VIBRACIONAL E LÉXICO DA JORNADA (EXPANDIDA)

Propósito

Criar um **dicionário oficial do FriendApp**, onde cada termo sensorial e vibracional tenha uma definição clara, poética e técnica.

Esse glossário serve para:

- Garantir consistência entre times de desenvolvimento, curadoria e comunidade.
- Orientar o usuário no contato com termos únicos do ecossistema.
- Registrar o **vocabulário vibracional** que diferencia o FriendApp de qualquer outra plataforma.

Entrada Sensorial

Toda jornada precisa de um idioma.


O **Glossário Vibracional** é a **língua sagrada do FriendApp**, onde cada palavra carrega energia, simbolismo e função real no sistema.

Assim, o usuário entende que não está apenas interagindo com tecnologia, mas **participando de uma cultura viva**.

Estrutura do Glossário

Termo Sensorial	Significado Vibracional	Uso Técnico no Sistema
Jornada Sensorial	Caminho de autodescoberta contínua	Linha do tempo vibracional (API /timeline)
Check-in Vibracional	Pausa consciente para registrar estado energético	expansao-service
Pulso de Expansão	Onda de energia após ação consciente	Logs → Feedback IA
Espaço Seguro	Grupo livre de julgamento, protegido por curadoria	comunidade-service
Frequência Dominante	Vibração emocional mais presente do usuário	Atualiza mapa + feed
Semente Cristalina	Recompensa simbólica por ato de empatia	Sistema de Selos
Aurah Kosmos	IA central que lê e responde energias	Motor principal de IA
Boas-vindas Vibracional	Ritual de entrada sensorial no app	Onboarding inicial
Regeneração Energética	Práticas automáticas em caso de queda vibracional	regeneracao-service
Mini-Oráculo Interno	Respostas profundas da IA em momentos existenciais	Submódulo de IA Aurah

Fluxo de Uso do Glossário

1. Usuário interage com função desconhecida (ex: recebe “Pulso de Expansão”).
2. Sistema identifica termo → dispara tooltip sensorial.
3. Tooltip exibe:
 - Definição poética
 - Função prática
 - Link para Glossário completo
4. Usuário pode abrir painel lateral “ Glossário Vibracional”.

Dados Monitorados

Tipo de Interação	Registro
Consultas ao glossário	Logado em glossario_logs
Termos mais pesquisados	Usado para priorizar explicações futuras
Atualizações IA	IA pode sugerir novos termos com base em padrões emergentes
Versões multilíngue	Gerado para expansão internacional

Pseudocódigo Simplificado


```
def mostrar_glossario(termo):
    definicao = buscar_definicao(termo)
    return {
        "poetico": definicao.poetico,
        "tecnico": definicao.tecnico,
        "uso": definicao.uso_sistema
    }
```

}


Integrações

Sistema Integrado	Uso do Glossário
IA Aurah Kosmos	Explica termos automaticamente em diálogos
Feed Sensorial	Exibe lembretes e definições nos cards
Onboarding	Apresenta termos básicos aos novatos
Gamificação	Desbloqueia novos termos conforme evolução
Tradução Multilíngue	Adapta definições para cada idioma

Exemplo Real

- Usuário recebe notificação: “Você ganhou uma **Semente Cristalina** .
- Não entende → clica no termo.
- Painel abre:
 - Definição Poética: “Um símbolo que nasce quando sua empatia floresce no coletivo.”
 - Função Técnica: “Registrado como recompensa simbólica em seu perfil.”
- Usuário sente pertencimento ao ver que sua ação tem **peso vibracional real**.

Fechamento

 O Glossário Vibracional é mais que documentação: é **a língua viva do FriendApp**.

Ele traduz tecnologia em símbolos, e símbolos em cultura.

Assim, cada usuário se sente parte de algo maior: uma comunidade com vocabulário próprio de expansão.

CAMADA 15 — ARQUITETURA GERAL DOS MICROSERVIÇOS (EXPANDIDA)

Propósito

Definir a **topologia técnica** que sustenta o bloco **Suporte ao Usuário + Comunidade + Jornada Sensorial**, garantindo **escala, segurança, privacidade (LGPD/GDPR)** e **baixa latência** nas respostas da IA Aurah Kosmos e nos fluxos vibracionais.

Entrada Sensorial

Imagine o FriendApp como um **organismo de luz**: cada órgão é um serviço, cada sinapse é um evento, cada batida é um **check-in vibracional**.

A arquitetura de microserviços permite que **cada parte pulse no seu ritmo**, sem perder a harmonia do corpo inteiro.

--

Visão Macro (Domínios & Serviços)

Domínio	Serviço	Responsabilidade	Exemplos de Endpoints
Identidade & Consentimento	auth-service	Autenticação/OIDC, tokens, MFA	POST /auth/login , POST /auth/refresh
	consent-service	Opt-in/opt-out, trilhas de consentimento	GET/POST /consent/{userId}
Jornada & Energia	checkin-service	Check-ins vibracionais, escala 0-10, texto/áudio	POST /checkins , GET /checkins/stats
	timeline-service	Linha do tempo, marcos/selo	GET /timeline , POST /timeline/events
	frequency-map-service	Mapa de Frequência (individual/coletivo)	GET /map , POST /map/update
IA & Curadoria	aurah-gateway	Orquestra a IA Aurah Kosmos (NLP/NLU, scoring)	POST /aurah/infer , GET /aurah/state
	curadoria-service	Score vibracional de conteúdos, filtros	POST /curadoria/eval
	recomendacao-service	Grupos, conteúdos, eventos afins	GET /recommendations
Suporte & Comunidade	suporte-service	Tickets, chat sensível, triagem (N1/N2/N3)	POST /support , POST /support/route
	guardioes-service	Programa Guardiões (formação, reputação)	POST /guardioes/apply , GET /guardioes/me
	mentores-service	Marketplace de Mentores (premium)	GET /mentores , POST /mentores/session
	comunidade-service	Grupos, posts, moderação	POST /groups , POST /posts
Conteúdo & Regeneração	conteudo-service	Biblioteca vibracional (áudio, vídeo, textos)	GET /content , POST /content/feedback
	regeneracao-service	Modo Alívio, protocolos automáticos	POST /regeneracao/ativar
Selos & Gamificação	selos-service	Recompensas simbólicas e marcos	POST /selos/grant , GET /selos/user
Observabilidade & Segurança	audit-service	Trilhas de auditoria criptografadas	POST /audit
	logging-service	Logs estruturados, correlação	POST /logs
	metrics-service	SLOs, painéis admin	GET /metrics

Topologia de Deploy (alto nível)

flowchart LR

```

Client(App/Web) → API_GW[API Gateway]
API_GW → Auth[auth-service]
API_GW → Aurah[aurah-gateway]
API_GW → Checkin[checkin-service]
API_GW → Curadoria[curadoria-service]
API_GW → Comunidade[comunidade-service]
API_GW → Suporte[suporte-service]

```

```

API_GW → Reco[recomendacao-service]
API_GW → Conteudo[conteudo-service]
API_GW → Reg[regeneracao-service]
Aurah ↔ Curadoria
Aurah ↔ Reco
Checkin →|events| Bus[(Event Bus)]
Comunidade →|events| Bus
Suporte →|events| Bus
Bus → Timeline[timeline-service]
Bus → Map[frequency-map-service]
Bus → Selos[selos-service]
subgraph Observabilidade
  Audit[audit-service]:::sec
  Logs[logging-service]:::sec
  Metrics[metrics-service]:::sec
end
classDef sec fill:#0b1220,stroke:#78D,color:#9cf

```

API Gateway com **rate-limit** por cliente/rota, JWT/OIDC, mTLS interno.

Padrões de Comunicação

- **Síncrono** (REST/gRPC): leitura/ações imediatas (ex.: `POST /checkins`).
- **Assíncrono** (Event Bus: Kafka/PubSub): *fan-out* de eventos sensoriais:
 - `event.checkin.created`
 - `event.support.routed`
 - `event.curadoria.flagged`
 - `event.regeneracao.started`
- **Sagas/Outbox Pattern** para consistência entre serviços (timeline/selos).

Bancos & Esquemas (resumo)

1) Relacional (PostgreSQL) — entidades canônicas

```

TABLE users(id PK, created_at, status, premium_level);
TABLE consents(user_id FK, key, value, version, timestamp);
TABLE groups(id PK, name, freq_min, owner_id, policy_json);
TABLE posts(id PK, group_id FK, user_id FK, content, score, created_at);
TABLE support_tickets(id PK, user_id, status, routed_to, severity, created_at);
TABLE mentors(id PK, profile, langs[], pricing_json, rating);

```

2) Time-series (Timescale/ClickHouse) — sinais vibracionais

```

TABLE checkins(user_id, ts, score NUMERIC, emotion, intensity, text_hash);
TABLE freq_map(region, ts, avg_score, active_users);

```

```
TABLE aurah_inferences(user_id, ts, confidence, labels[], features_json);
```

3) Documental (Mongo/S3) — mídia & payloads IA

- `content_assets` (áudio 432/528Hz, vídeos),
- `aurah_contexts` (janelas de contexto, embeddings),
- `regeneracao_protocols` (playlists/rotas).

4) Criptografia sensível (KMS/HSM)

- Campos PII (nome, e-mail), logs emocionais, *session tokens*.

Segurança & Privacidade (LGPD/GDPR-ready)

- **Consentimento granular** (Camada Extra 01): `consent-service` como *source of truth*.
- **Minimização de dados**: só coletar o necessário para a finalidade.
- **Base legal**: Execução de contrato (uso do app) + Consentimento para sinais emocionais.
- **Direitos do titular**: exportar/retificar/apagar (`/privacy/dsr`).
- **Retenção**:
 - Check-ins: 18 meses (padrão) → anonimização após.
 - Logs emocionais: 90 dias (auditoria ética).
- **Criptografia**: em trânsito (TLS 1.3), em repouso (AES-256/KMS).
- **Segredos**: Vault/KMS, rotação automática.
- **PII Boundary**: serviços “friamente” isolados por *network policies*.

SLOs, Escala & Resiliência

Área	SLO	Estratégia
Latência IA (inferência Aurah)	p95 ≤ 600 ms	<i>Warm pools</i> , batch features
Check-ins ingest	p95 ≤ 150 ms	gRPC + TSDB <i>append</i>
Suporte N1	≤ 2 s	<i>Autoscaling</i> + fila priorizada
Disponibilidade	≥ 99.9%	Multi-zona + <i>self-heal</i>
Perda de dados	RPO ≤ 5 min	WAL + réplica multi-região
Recuperação	RTO ≤ 30 min	<i>Runbooks</i> + automação DR

Observabilidade: OpenTelemetry (traces/logs/metrics), *error budgets* por serviço, *golden signals* (latência, tráfego, erros, saturação).

Feature Flags, A/B e Rollouts

- **Flags** por coorte vibracional (ex.: grupos em queda recebem UX mais leve).
- **A/B**: recomendação de conteúdos vs. comunidades.
- **Rollout**: canário + *progressive delivery* (1% → 25% → 100%).
- **Kill-switch** imediato para modelos IA.

Orquestrações Críticas (Sagas)

Exemplo: Check-in → Recomendação → Selos

```
sequenceDiagram
    participant App
    participant Checkin as checkin-service
    participant Bus as Event Bus
    participant Reco as recomendacao-service
    participant Timeline
    participant Selos

    App->>Checkin: POST /checkins
    Checkin->>Bus: publish event.checkin.created
    Bus->>Reco: consume event
    Reco->>App: cards recomendados
    Bus->>Timeline: upsert user timeline
    Reco->>Bus: publish event.reco.accepted (if user accepts)
    Bus->>Selos: grant "Pulso Cósmico"
```

Fluxos de Fallback & Degradação Elegante

- **IA Aurah indisponível** → respostas “baseadas em regras” + copy sensível.
- **Curadoria fora** → feed restrito ao círculo seguro (últimos conteúdos validados).
- **Mentores offline** → fila de guarda e reagendamento automático.
- **Eventos quebrados** → cache regional + banner de transparência.

CI/CD & Qualidade

- Pipelines GitOps (Build → Scan SAST/DAST → Teste → Deploy canário).
- Testes: unitário, contrato (Pact), carga, caos (latência/partição).
- *Policy as Code*: verificação de schemas, *lint* de OpenAPI, RBAC.

Matriz de Acesso (RBAC)

Papel	Permissões
Usuário	check-ins, comunidades, suporte N1
Guardião	moderação limitada, responder N2, ver métricas agregadas
Mentor (Premium)	sessões 1-a-1, acesso controlado a contexto com consentimento explícito
Admin Ético	auditoria/observabilidade; sem acesso a conteúdo privado bruto

Exemplos de Contratos (OpenAPI high-level)

```
POST /checkins:
requestBody: { userId, score, emotion, text?, audioRef? }
```

```
responses: { 201, 400 }
```

POST /aurah/infer:

```
requestBody: { userId, input, context?, consentToken }  
responses: { 200: {confidence, labels[], actionHints[]} }
```

POST /regeneracao/ativar:

```
requestBody: { userId, reason, protocolId? }  
responses: { 202 }
```

Integrações Internas (eventos principais)

Evento	Emissor	Consumidores	Efeito
event.checkin.created	checkin-service	timeline , recomendacao , frequency-map , selos	Atualiza jornada, gera cards e selos
event.support.routed	suporte-service	guardioes , mentores , audit	Triagem N1/N2/N3
event.curadoria.flagged	curadoria-service	comunidade , audit	Moderação educativa
event.regeneracao.started	regeneracao-service	timeline , metrics	Marca ciclo de cuidado


DR, Backups & Data Lifecycle

- **Backups:** diários (full) + hora-a-hora (incremental).
- **DR:** *pilot-light* em região secundária, *infrastructure as code*.
- **Lifecycle:** arquivamento/anonimização automática conforme política.

Pseudocódigo de Orquestração (Check-in → IA → Feed)

```
def processar_checkin(payload):  
    assert consent_ok(payload.user_id, "emotional_processing")  
    save_checkin(payload)  
    publish("event.checkin.created", payload)  
  
@subscribe("event.checkin.created")  
def gerar_insights(event):  
    infer = aurah.infer(event.user_id, event)  
    update_timeline(event.user_id, infer)  
    recos = recomendacao.gerar(event.user_id, infer)  
    push_cards(event.user_id, recos)
```

Fechamento

 Esta arquitetura garante que o FriendApp **respire em micro-pulsos**, com **IA confiável**, **segurança ética** e **escala global** — sem perder o toque humano e o cuidado vibracional.

Com essa engrenagem, avançamos para as próximas camadas técnicas sabendo que o corpo inteiro está firme e luminoso.

CAMADA 16 — TABELA DE DADOS MONITORADOS & TELEMETRIA VIBRACIONAL (EXPANDIDA)

Propósito

Definir **quais dados** o bloco **Suporte + Comunidade + Jornada Sensorial** coleta, **como** esses dados são processados e **com que finalidade vibracional e técnica** são usados — garantindo **privacidade, ética, LGPD/GDPR**, observabilidade e capacidade analítica para tomada de decisão da IA Aurah Kosmos e dos módulos humanos (Guardiões/Mentores).

Entrada Sensorial

Tudo que pulsa deixa rastro. No FriendApp, cada rastro é tratado como **sopro sagrado**: coletado com **consentimento**, armazenado com **sigilo**, e utilizado com **intenção terapêutica**.

Telemetria aqui não é vigilância: é **escuta ética** da jornada, para que possamos **proteger, regenerar e expandir**.

Mapa de Dados (o que coletamos, por quê, e por quanto tempo)

Categoria	Campo (exemplos)	Fonte	Finalidade Vibracional/Técnica	Base Legal	Retenção
Perfil Básico	<code>user_id</code> , país, idioma	Cadastro	Localização de clusters culturais (recomendações)	Execução do contrato	Enquanto ativo
Consentimentos	chaves de opt-in/out	Painel de Privacidade	Limitar/autorizar leitura emocional e ações proativas	Consentimento explícito	Permanente (histórico)
Check-ins	score 0–10, emoção, texto/áudio hash	Jornada	Curva vibracional, gatilhos de regeneração	Consentimento	18 meses (anonimização após)
Comunidades	grupos, posts, interações	Comunidade	Curadoria, prevenção de toxicidade	Execução do contrato	12 meses
Suporte	tipo, severidade, rota (N1/N2/N3)	Canal de Suporte	Triagem, auditoria ética	Interesse legítimo + consent.	90 dias (conteúdo bruto), metadados 12m
IA Inferences	labels, confidence, action_hints	Aurah	Explicabilidade + tuning	Interesse legítimo	12 meses
Conteúdos Regenerativos	play, completion, feedback	Biblioteca	Eficácia terapêutica, recomendações	Execução do contrato	12 meses

Categoria	Campo (exemplos)	Fonte	Finalidade Vibracional/Técnica	Base Legal	Retenção
Recompensas/Selos	símbolo, motivo, data	Gamificação	Reconhecimento e engajamento consciente	Execução do contrato	Enquanto ativo
Eventos	RSVP, presença, feedback	Eventos	Ajuste de recomendações presenciais	Execução do contrato	12 meses
Observabilidade	métricas, traces, logs	Infra	Confiabilidade, SLOs, segurança	Interesse legítimo	30–90 dias (logs)

Nota ética: conteúdo sensível (texto/áudio do suporte, mensagens privadas) não é usado para treinar modelos sem consentimento específico e nunca em formato identificável.

Esquemas & Armazenamento (resumo técnico)

Relacional (PostgreSQL)

```
TABLE consents(
  user_id UUID, key TEXT, value BOOLEAN, version INT, ts TIMESTAMP, PRIMARY KEY(user_id, key, version)
);

TABLE checkins(
  user_id UUID, ts TIMESTAMP, score NUMERIC(3,1), emotion TEXT, text_hash TEXT, audio_ref TEXT NULL
);

TABLE community_posts(
  id UUID PK, group_id UUID, user_id UUID, created_at TIMESTAMP,
  score_vibr NUMERIC(3,1), status TEXT CHECK(status IN ('ok','partial','hidden'))
);

TABLE support_tickets(
  id UUID PK, user_id UUID, severity INT, routed_to TEXT, status TEXT, created_at TIMESTAMP
);

TABLE rewards(
  id UUID PK, user_id UUID, symbol TEXT, reason TEXT, ts TIMESTAMP
);
```

Time-series (ClickHouse/TimescaleDB)

```
TABLE telemetry_checkins(user_id UUID, ts DateTime, score Float32, trend Float32, shard UInt16);
TABLE aurah_infer(user_id UUID, ts DateTime, confidence Float32, labels Array(String));
TABLE content_effect(ts DateTime, user_id UUID, content_id UUID, delta_score Float32);
```

Documental (Mongo/S3)

- `aurah_contexts` : janelas de contexto, *explanations*, *action_hints*.
- `regeneracao_protocols` : playlists, scripts, assets.
- `glossario` : versões do léxico e tooltips.

Criptografia/KMS

- Campos sensíveis (PII, inferências emocionais) com **field-level encryption**.
- Segredos via **Vault/KMS** com rotação automática.

Pipeline de Telemetria (coleta → processamento → ação)

flowchart LR

```

App → |events| Edge[Edge Collector]
Edge → Bus[(Event Bus)]
Bus → ETL[Stream Processor]
ETL → TSDB[(Time-series DB)]
ETL → Feat[Feature Store]
Feat → Aurah[aurah-gateway]
Aurah → Act[Action Engine]
Act → Reco[recomendacao-service]
Act → Regen[regeneracao-service]
Act → Curad[curadoria-service]
ETL → OLAP[(Warehouse/OLAP)]
OLAP → Dash[Dashboards/Admin]

```

- **Edge Collector**: valida consentimento, enfileira com *backpressure*.
- **Stream Processor** (Flink/Spark/Kinesis): limpa, agrega, calcula tendências.
- **Feature Store**: mantém *features* online/offline para a IA.
- **Action Engine**: aplica regras + *policy* de ética e confiança antes de agir.

Métricas Vibracionais & Indicadores Operacionais

Indicador	Descrição	Cálculo (exemplo)	Uso
Índice de Expansão	variação positiva média em 7/30 dias	<code>Δ score_7d vs 30d</code>	saúde da jornada
Taxa de Regeneração	% usuários que melhoram ≤48h pós-protocolo	<code>improved_48h / total</code>	eficácia de cuidado
Queda Coletiva	% de check-ins < limiar em 24h	<code>count(score<3)/total</code>	alerta de campo
Afinidade de Comunidade	match entre usuário e grupo	cosine entre vetores	curadoria
Eficácia de Conteúdo	Δ médio pós-consumo	<code>avg(delta_score)</code>	biblioteca vibracional
Escalonamentos Éticos	casos N3 (humano) por 1k interações	contagem normalizada	dimensionamento equipe

Cálculos-Chave (pseudocódigo)

Tendência Vibracional (usuário)

```
def tendencia_7_30(scores):
    m7 = sum(scores[-7:]) / max(1, len(scores[-7:]))
    m30 = sum(scores[-30:]) / max(1, len(scores[-30:]))
    return (m7 - m30) / max(0.1, m30)
```

Eficácia de Regeneração (conteúdo/prática)

```
def eficacia_regeneracao(events):
    # events: [(ts, score_before, score_after)]
    wins = [1 for e in events if (e.score_after - e.score_before) >= 0.5]
    return sum(wins) / max(1, len(events))
```

Afinidade de Comunidade (vetor semântico + vibração)

```
def afinidade(usuario_vec, grupo_vec, freq_user, freq_min):
    base = cosine(usuario_vec, grupo_vec)
    gate = 0 if freq_user < freq_min else 1
    return base * gate
```

Governança de Dados & Privacidade

- **Consent by design:** toda leitura emocional exige **chave de consentimento** (`consent.emotional_processing == true`).
- **Propósito específico:** cada pipeline declara finalidade (recomendar, regenerar, curar).
- **Acesso:** RBAC estrito; **Admin Ético** audita, mas **não** lê conteúdo privado bruto.
- **Direitos do Usuário:** exportação/retificação/anonimização (`/privacy/dsr`).
- **Logs Sensíveis:** criptografados, **90 dias** de retenção.
- **Anonimização:** *tokenization* para análises agregadas (OLAP).

Alertas & Painéis (Observabilidade Sensorial)

Alerta	Condição	Ação
Queda Coletiva	check-ins <3 > 12% em 24h	ativar banner de acolhimento + campanha “Flutua”
Tóxico em Grupo	<code>curadoria.flagged</code> > X/h	reduzir alcance, acionar Guardiões
Regeneração Ineficaz	eficácia < 30% por 48h	revisar protocolo, A/B de novo conteúdo
Fila de Suporte N3	SLA > 3 min	chamar plantão Guardiões/Mentores
Falha de Consent	eventos bloqueados ↑	auditar consent-service

Dashboards (Grafana/Metabase):

- Jornada & Expansão (curvas 7/30/90 dias).
- Regeneração & Conteúdos (eficácia por protocolo).
- Comunidades & Saúde (afinidade, flags, guardiões ativos).

- Suporte & Ética (N1/N2/N3, escalonamentos, SLA).

Eventos Canônicos (para integração entre serviços)

Evento	Payload (resumo)	Consumidores	Efeito
<code>event.checkin.created</code>	<code>{user, ts, score, emotion}</code>	timeline, recomendacao, freq_map, selos	Atualiza jornada e sugestões
<code>event.support.opened</code>	<code>{ticket_id, user, severity}</code>	guardioes, mentores, audit	Triagem e rota
<code>event.curadoria.flagged</code>	<code>{post_id, user, reason}</code>	comunidade, audit	Moderação educativa
<code>event.regeneracao.started</code>	<code>{user, protocol_id}</code>	timeline, metrics	Marca ciclo de cuidado
<code>event.reward.granted</code>	<code>{user, symbol, reason}</code>	timeline, feed	Reconhecimento simbólico

Testes & Qualidade de Dados

- **Contracts:** schemas de eventos validados (Avro/JSON Schema).
- **Data Health:** *freshness*, completude, range checks (Great Expectations).
- **Drift:** monitorar distribuição de scores/labels de IA (alertas).
- **Reprodutibilidade:** *feature store* com *versioning* (online/offline parity).

Retenção, Anonimização & Lifecycle

- **Check-ins:** 18 meses → **anonimiza** (hash de `user_id`, remoção de texto).
- **Logs de suporte:** 90 dias → **delete**; metadados mantidos 12m.
- **Inferências da IA:** 12 meses → agregação/anon.
- **Direito de esquecimento:** *cascade delete* + invalidar chaves no KMS.

Exemplo de Telemetria (ingestão → ação)

```
def on_checkin(event):
    if not consent_ok(event.user_id, "emotional_processing"):
        return
    save_tsdb(event)
    trend = tendencia_7_30(load_scores(event.user_id))
    infer = aurah.infer(event.user_id, {"score": event.score, "trend": trend})
    if infer.action == "regenerar":
        start_protocol(event.user_id, infer.protocol_id)
    push_recommendations(event.user_id, infer.recos)
```

✨ Fechamento

📌 Esta camada garante **clareza total** sobre **que dados existem, por que existem, por quanto tempo existem** — e **como** se tornam **cuidado real** dentro do FriendApp.

Telemetria aqui é **ética aplicada**: tudo o que medimos é para **proteger, regenerar e expandir** o usuário e o campo coletivo.

Com esta base sólida, seguimos para a engrenagem de vulnerabilidade e detecção sensível — onde a matemática encontra a compaixão.

CAMADA 17 — MATRIZ DE VULNERABILIDADE & SCORE ÉTICO DE RISCO EMOCIONAL (EXPANDIDA)

Propósito

Traduzir a ideia de “**cuidado vibracional proativo**” em **métrica objetiva, transparente e ética**, para que a IA Aurah Kosmos consiga detectar estados de vulnerabilidade sem invadir a privacidade do usuário.

Essa matriz garante **previsibilidade, consistência** e, sobretudo, **respeito ao consentimento**.

Entrada Sensorial

O maior desafio da IA vibracional é sentir **sem invadir**.


A vulnerabilidade precisa ser **detectada com clareza matemática**, mas também **respeitada como escolha humana**.

Por isso, o FriendApp adota um modelo de **Matriz de Vulnerabilidade**:

- Escala **0–10** de risco vibracional.
- Baseada apenas em **sinais consensuais e públicos**.
- Ativa protocolos de cuidado apenas se **opt-in do usuário estiver ativo**.

Sinais Considerados (Opt-in + Públicos)

Indicador	Fonte	Descrição	Peso (w)
Inatividade Social	Check-ins, comunidades	Dias sem interações ou registros	0.35
Feedbacks Negativos	Pós-eventos, suporte	Usuário relata experiência ruim	0.25
Linguagem Pública de Colapso	Posts em comunidades, feed	Palavras/expressões de baixa vibração	0.25
Redução de Interações Positivas	Gamificação, recompensas	Queda brusca de ações elevadas	0.15

 **Nunca são usados:** mensagens privadas, áudio/texto confidencial de suporte, ou dados sem consentimento.

Fórmula do Score de Vulnerabilidade

$$\text{ScoreVulnerabilidade} = (w1 * \text{InatividadeSocial}) + (w2 * \text{FeedbacksNegativos}) + (w3 * \text{LinguagemPublicaColapso}) - (w4 * \text{Interac\c{o}~esPositivasRecentes})$$
$$\text{Score_Vulnerabilidade} = (w1 * \text{Inatividade_Social}) + (w2 * \text{Feedbacks_Negativos}) + (w3 * \text{Linguagem_Publica_Colapso}) - (w4 * \text{Intera\c{c}~es_Positivas_Recentes})$$

$\text{ScoreVulnerabilidade} = (w1 * \text{InatividadeSocial}) + (w2 * \text{FeedbacksNegativos}) + (w3 * \text{LinguagemPublicaColapso}) - (w4 * \text{Interac\c{oesPositivasRecentes})$

- **0-2** → Estável
- **3-5** → Atenção
- **6-8** → Risco vibracional
- **9-10** → Vulnerabilidade crítica (aciona humano, se opt-in ativo)

Pseudocódigo de Cálculo

```
def calcular_vulnerabilidade(user):
    inatividade = dias_sem_interacao(user)
    feedback_neg = count_feedbacks_negativos(user)
    linguagem_colapso = detectar_linguagem(user.posts_publicos)
    interacoes_pos = count_interacoes_positivas(user, ultimos=7)

    score = (0.35 * inatividade) + (0.25 * feedback_neg) \
        + (0.25 * linguagem_colapso) - (0.15 * interacoes_pos)

    return max(0, min(score, 10))
```

Exemplos Práticos

- **Exemplo 1 — Usuário Estável**
 - Check-ins diários, comunidades ativas, posts positivos.
 - Score = 1,8 → Estável.
- **Exemplo 2 — Usuário em Atenção**
 - 10 dias sem check-ins, 2 feedbacks negativos em eventos.
 - Score = 4,2 → Atenção.
- **Exemplo 3 — Usuário em Risco**
 - Posts públicos: "Nada faz sentido, estou cansado".
 - Check-ins raros, queda de interações positivas.
 - Score = 7,5 → Risco vibracional → Sugestão de prática + grupo de apoio.
- **Exemplo 4 — Usuário em Crise Crítica**
 - 20 dias sem interações, posts públicos de colapso, feedbacks negativos consecutivos.
 - Score = 9,2 → Vulnerabilidade crítica → Escalonamento humano (se opt-in ativo).

Integrações da Matriz

Sistema	Uso do Score
Aurah Kosmos	Ajusta respostas e práticas sugeridas
Suporte Inteligente	Define se caso deve ser escalonado para Guardião/Mentor

Sistema	Uso do Score
Regeneração Automática	Ativa protocolos suaves de cuidado (ex.: Modo Alívio)
Painel Admin Ético	Mostra apenas estatísticas agregadas, nunca casos individuais
Gamificação & Selos	Pode recompensar quando usuário se recupera de quedas

Ética & Segurança

- O usuário deve ativar manualmente o **opt-in** para receber suporte proativo.
- Sem consentimento, a IA apenas **observa estatísticas coletivas**.
- Logs de score são **anonimizados após 12 meses**.
- Guardiões e Mentores só veem o score se o usuário **autorizar durante a sessão**.
- O sistema nunca usa **mensagens privadas** para calcular vulnerabilidade.

✨ Fechamento

📌 A Matriz de Vulnerabilidade traduz poesia em matemática:

sente sem invadir, cuida sem vigiar, acolhe sem punir.

Ela é o **escudo ético da IA Aurah Kosmos**, garantindo que cada intervenção seja legítima, consciente e escolhida pelo próprio usuário.

CAMADA 18 — ESTRUTURA DE LOGS CRIPTOGRAFADOS & AUDITORIA ÉTICA (EXPANDIDA)

Propósito

Definir a forma como todas as interações emocionais, comunitárias e de suporte são **registradas, protegidas e auditadas** dentro do FriendApp.

O objetivo é equilibrar **observabilidade técnica** com **privacidade vibracional**, garantindo que os logs sirvam para evolução, mas nunca para vigilância.

Entrada Sensorial

Cada palavra, gesto ou check-in deixa uma marca digital.

No FriendApp, essa marca não é explorada: é **guardada em cofres de luz**, criptografada, acessível apenas para fins de **cura e segurança coletiva**.

Os logs aqui não são controle: são **memória ética** do campo.

Estrutura dos Logs

Categoria	Evento Monitorado	Registro no Log	Finalidade
Check-ins	Escala emocional + emoção declarada	<code>user_id (hash), ts, score, emotion</code>	Construir curva vibracional
IA Suporte	Interações com Aurah	<code>intent, resposta, confiança</code>	Avaliar performance da IA

Categoria	Evento Monitorado	Registro no Log	Finalidade
Comunidade	Postagens e comentários públicos	post_id, score_vibracional, status	Moderação educativa
Escalonamentos	Rota N1/N2/N3 (IA, Guardião, Mentor)	ticket_id, nível, ts	Garantir consistência ética
Protocolos de Regeneração	Ativação de Modo Alívio ou práticas	user_id (hash), protocolo_id, ts	Auditoria terapêutica
Recompensas	Conquistas simbólicas	user_id (hash), reward_symbol, ts	Transparência no reconhecimento

Fluxo Técnico de Registro

1. Usuário realiza ação (check-in, post, suporte).
2. Evento é enviado ao `logging-service`.
3. Dados são normalizados, com:
 - user_id → hash+salt
 - conteúdo → texto tokenizado (sem armazenar bruto)
4. Registro salvo em 2 camadas:
 - ****Time-series DB**** (tendência, métricas)
 - ****Cold Storage criptografado**** (auditoria)
5. Logs críticos → enviados ao `audit-service`.

Modelo de Log (JSON)

```
{
  "log_id": "uuid",
  "ts": "2025-09-13T18:45:00Z",
  "user_hash": "sha512:abc123...",
  "module": "checkin-service",
  "event_type": "checkin_created",
  "payload": {
    "score": 4.2,
    "emotion": "tristeza",
    "trend": "queda"
  },
  "consent_token": "uuid"
}
```

Camadas de Criptografia & Segurança

- **Criptografia em repouso:** AES-256 com chaves rotativas (KMS).
- **Criptografia em trânsito:** TLS 1.3 + mTLS entre microsserviços.
- **PII Boundary:** separação de identificadores pessoais dos dados emocionais.
- **Consent Token:** cada log exige chave de consentimento válida.

- **Anonimização periódica:** após 90 dias, conteúdo bruto é descartado; metadados permanecem anonimizados.

Auditoria Ética

- **Admin Ético:** papel específico com acesso apenas a estatísticas agregadas, nunca a conteúdo bruto.
- **Relatórios Trimestrais:** métricas sobre:
 - Qtd. de logs processados
 - Qtd. de escalonamentos humanos
 - Eficácia de protocolos de regeneração
- **Auditoria Externa:** possibilidade de revisão por conselho independente (comunidade + especialistas).


Observabilidade & Métricas

Métrica	Descrição	Uso
Taxa de logs válidos	% eventos com consent_token aceito	Garantir respeito ao opt-in
Tempo médio de retenção	Quantos dias até anonimização	Cumprir política LGPD/GDPR
Escalonamentos éticos	Casos críticos tratados por humanos	Monitorar carga dos Guardiões/Mentores
Falhas de log	Eventos perdidos/descartados	Ajustar pipeline técnico

Exemplo Real

- Usuário Mariana faz check-in às 23h relatando “cansaço extremo”.
- Evento logado:
 - `score = 2,8`
 - `trend = queda`
- IA Aurah sugere prática → protocolo regenerativo ativado.
- Log mostra:
 - `event.regeneracao.started`
 - `user_hash: xxx`
- Após 48h, Mariana faz novo check-in: score 4,5.
- Auditoria confirma eficácia do protocolo.

Fechamento

 Os logs criptografados e a auditoria ética transformam dados em **confiança**.

Aqui, cada registro é **memória vibracional protegida**, usada apenas para que a plataforma saiba **quando e como cuidar melhor do usuário**.

CAMADA 19 — PAINEL FRONTEND: AJUDA, JORNADA SENSORIAL E COMUNIDADES SUGERIDAS (EXPANDIDA)

Propósito

Projetar a **interface visível ao usuário** para acesso ao suporte, acompanhamento da jornada e entrada em comunidades sugeridas.

Este painel é o **espelho vivo** do ecossistema vibracional, traduzindo dados e insights em **experiência sensorial interativa**.

Entrada Sensorial

O painel frontend é a **porta de entrada diária** do usuário no FriendApp.

Aqui ele vê:

- Sua evolução energética.
- Recomendações da IA Aurah.
- Sugestões de comunidades afins.
- Botão de acesso imediato ao suporte.

Nada aqui é estático: a tela **muda conforme a frequência do usuário**, pulsando junto com ele.

Estrutura do Painel

Módulo	Função	Integrações
Área de Ajuda	FAQ sensorial, botão “Falar com a Aurah”	Suporte-Service + IA Aurah
Linha da Jornada	Exibe evolução vibracional (curva + selos)	Check-in + Timeline-Service
Comunidades Sugeridas	Cards interativos de grupos afins	Curadoria + Recomendação-Service
Feedback Emocional	Usuário avalia suporte/interação	Logs + Regeneração-Service
Alertas Positivos	Frases e sons de incentivo	Biblioteca Vibracional + IA
Sugestões Dinâmicas	Conteúdos, práticas, eventos	Aurah + Recomendação-Service

Fluxo Operacional

1. Usuário abre o painel.
2. API Gateway chama:
 - /timeline → curva vibracional
 - /recommendations → grupos/eventos
 - /support/faq → FAQ sensorial
3. Aurah avalia estado atual do usuário.
4. Frontend renderiza:
 - Curva com cores dinâmicas
 - Cards com comunidades afins
 - Botão de suporte pulsante se score <3

5. Logs são enviados ao metrics-service para análise de uso.

Design Sensorial

- **Cores:** mudam conforme estado vibracional (verde = expansão, azul = estável, vermelho = queda).
- **Animações:** curvas da jornada respiram suavemente.
- **Áudio:** tons em 432Hz/528Hz ao interagir com cards de suporte.
- **Tipografia:** frases curtas, poéticas, nunca técnicas.

Dados Monitorados

Dado	Uso
Cliques em cards de comunidade	Refinar recomendações
Consultas ao FAQ	Melhorar respostas automáticas
Solicitações de suporte	Ajustar fila de Guardiões/Mentores
Interações com gráficos	Medir engajamento da jornada
Feedback emocional pós-suporte	Calibrar IA Aurah e protocolos

Exemplo JSON de Renderização

```
{
  "timeline": {
    "trend": "queda",
    "scores": [6.1, 5.8, 4.9, 4.2],
    "color": "red"
  },
  "recommendations": [
    {"type": "group", "name": "Círculo de Acolhimento Noturno"},
    {"type": "event", "name": "Meditação da Aurora"}
  ],
  "alerts": [
    {"text": "Sua energia pede cuidado hoje. Respire com Aurah.", "tone": 432}
  ]
}
```

Exemplo Real

- Usuário abre painel às 23h:
 - Curva vibracional mostra queda.
 - Painel exibe card: *"Sua energia pede acolhimento. Conheça o grupo Resenha Noturna 🌙"*.
 - Botão de suporte aparece pulsando suavemente.
 - Ao clicar, IA Aurah responde: *"Estou aqui. Vamos cuidar da sua energia."*

✨ Fechamento

📌 O painel frontend é mais que interface: é **espelho da jornada**.

Nele, o usuário vê sua alma digital refletida, sempre acompanhada, cuidada e guiada pelo ecossistema FriendApp.

CAMADA 20 — SISTEMA DE CHECK-INS VIBRACIONAIS: LEITURA, SUGESTÃO E EVOLUÇÃO CONTÍNUA (EXPANDIDA)

🎯 Propósito

Estabelecer o **pulso vital** da jornada do usuário no FriendApp: os check-ins vibracionais.

Este sistema transforma simples registros de humor/emoção em **curvas de evolução energética**, acionando recomendações, regenerações e selos de reconhecimento.

Entrada Sensorial

No FriendApp, o check-in não é burocracia.

É um **ritual de escuta interior**:

- O usuário pausa.
- Respira.
- Registra como está.

E, a partir desse gesto simples, toda a engrenagem técnica e vibracional se recalibra.

Estrutura do Sistema

Elemento	Função	Integrações
Registro de Estado	Escala vibracional (0–10) + emoção declarada	Checkin-Service
Input Sensorial	Emojis, texto ou áudio opcional	NLP + IA Aurah
Linha do Tempo	Atualiza curva pessoal	Timeline-Service
Insights da IA	Sugestões de práticas/grupos	Aurah Kosmos
Recompensas	Pulso Cósmico, Selos de Consistência	Selos-Service
Protocolos de Regeneração	Ativados em quedas	Regeneracao-Service

Fluxo Operacional

1. Usuário abre aba "Meu Estado Vibracional".
2. Seleciona emoção atual (ex: paz, ansiedade, cansaço).
3. Define intensidade em slider (0–10).
4. Opcional: adiciona texto ou grava áudio.
5. Dados enviados → API /checkin/vibracional.
6. IA Aurah processa:

- Score atual
 - Tendência (expansão/queda/estável)
 - Insights para próximo passo
7. Linha do tempo atualizada.
 8. Recomendações personalizadas exibidas.
 9. Se consistência ≥ 7 dias \rightarrow Pulso Cósmico atribuído.

Dados Capturados

Dado	Finalidade
Escala vibracional (0–10)	Base da curva de evolução
Emoção declarada	Contextualização semântica
Emojis/símbolos	Entrada rápida de vibração
Texto/áudio opcional	Enriquecimento emocional via NLP
Tempo de resposta	Detectar hesitação/bloqueio
Frequência de check-ins	Calcular consistência

Algoritmo de Tendência

```
def tendencia_vibracional(checkins):
    media_7d = sum(checkins[-7:]) / max(1, len(checkins[-7:]))
    media_30d = sum(checkins[-30:]) / max(1, len(checkins[-30:]))
    delta = (media_7d - media_30d) / max(0.1, media_30d)
    if delta > 0.1:
        return "expansao"
    elif delta < -0.05:
        return "queda"
    else:
        return "estavel"
```

Integrações

Sistema	Função
IA Aurah Kosmos	Interpreta emoção, gera insights
Feed Sensorial	Ajusta conteúdos exibidos
Mapa de Frequência	Atualiza clusters coletivos
Gamificação/Selos	Recompensa constância
Sistema de Regeneração	Ativa protocolos em quedas críticas

Exemplo Real

- João faz 5 check-ins seguidos em uma semana.
 - Score médio cai de 6,2 \rightarrow 4,0.

- IA detecta queda consistente.
- Ações disparadas:
 - Linha do tempo exibe curva vermelha.
 - Aurah envia insight: *"Sua energia pede cuidado. Que tal experimentar Respiração Solar 432Hz?"*
 - Recomenda grupo "Acolhimento Noturno".
 - Se queda persistir → protocolo de regeneração ativado.

✨ Fechamento

📌 O sistema de check-ins é o **pulmão do FriendApp**: respira junto com o usuário, expira cuidado, inspira expansão.

Cada registro é mais que dado: é **ato consciente de autodescoberta**.

🌀 CAMADA 21 — SISTEMA DE LOGS CRIPTOGRAFADOS DE INTERAÇÕES EMOCIONAIS (EXPANDIDA)

🎯 Propósito

Registrar de forma **segura, ética e auditável** todas as interações emocionais, check-ins e eventos de suporte, garantindo rastreabilidade para melhoria contínua do sistema sem comprometer a privacidade do usuário.

🖥️ Entrada Sensorial

Cada emoção expressa no FriendApp deixa uma marca digital.

Mas essa marca não é propriedade da plataforma: é **memória vibracional do usuário**, protegida por camadas de criptografia e acessível apenas para **cuidado e auditoria ética**.

O sistema de logs é o **diário invisível da jornada coletiva**, garantindo que nada se perca e que tudo seja tratado com respeito.

🧩 Estrutura dos Logs

Categoria	Evento	Registro no Log	Finalidade
Check-ins	Registro vibracional	user_id (hash), score, emoção, ts	Curva de evolução
IA Suporte	Interação com Aurah	intenção detectada, resposta, confiança	Calibrar modelo
Comunidade	Postagens públicas	post_id, score, status	Moderação curativa
Regeneração	Ativação de protocolos	user_id (hash), protocolo_id, ts	Auditoria terapêutica
Escalonamento	Triagem N1/N2/N3	ticket_id, rota, ts	Controle de suporte humano
Recompensas	Selos e símbolos	reward_id, razão, ts	Transparência de reconhecimento

Fluxo Operacional

1. Usuário realiza ação (check-in, post, suporte).
2. Evento disparado → logging-service.
3. Dados sensíveis são:
 - Criptografados (AES-256)
 - Identificadores → hash+salt
 - Conteúdo → tokenizado (sem texto bruto)
4. Registro salvo em:
 - TSDB (tendências)
 - Storage frio (auditoria)
5. Logs críticos → audit-service (auditoria ética).

Modelo de Log (JSON)

```
{
  "log_id": "uuid",
  "ts": "2025-09-13T19:05:00Z",
  "user_hash": "sha512:xyz...",
  "module": "checkin-service",
  "event_type": "checkin_created",
  "payload": {
    "score": 3.8,
    "emotion": "ansiedade",
    "trend": "queda"
  },
  "consent_token": "uuid"
}
```

Segurança & Privacidade

- **Criptografia de ponta a ponta:** AES-256 em repouso, TLS 1.3 em trânsito.
- **Consentimento obrigatório:** todos os logs exigem `consent_token`.
- **Anonimização:** após 90 dias, conteúdo bruto descartado.
- **Retenção:** metadados ficam 12 meses para estatísticas.
- **Acesso:** só Admin Ético tem permissão — e apenas em nível agregado.
- **Proibição absoluta:** sem uso para anúncios ou perfis comerciais.

Auditoria Ética

- **Trilhas de auditoria:** cada log tem registro de origem, finalidade e uso.

- **Painel de Métricas Éticas:** monitora quantos escalonamentos ocorreram, quantos protocolos regenerativos foram eficazes.
- **Auditoria Externa:** conselho independente pode revisar logs anonimizados para validar confiança.

Métricas do Sistema

Métrica	Objetivo
Taxa de logs válidos (com consentimento)	Garantir respeito ao opt-in
Tempo médio até anonimização	Cumprir LGPD/GDPR
Falhas de log	Detectar problemas técnicos
Escalonamentos críticos	Dimensionar suporte humano

Exemplo Real

- Usuário faz check-in com score 2, relatando “cansaço extremo”.
- Log registrado → `event.checkin.created` com score = 2, trend = queda.
- IA ativa protocolo regenerativo → logado como `event.regeneracao.started`.
- Após 48h, novo check-in sobe score para 4,5 → log registrado como evolução positiva.
- Auditoria confirma eficácia do protocolo.

Fechamento

 O sistema de logs criptografados é o **guardião silencioso da memória vibracional**.

Aqui, dados não viram exploração — viram **proteção, cura e transparência**.

CAMADA 22 — MECANISMOS DE REGENERAÇÃO VIBRACIONAL (EXPANDIDA)

Propósito

Ativar protocolos automáticos e conscientes de **cuidado energético** quando o sistema detecta **quedas vibracionais** ou padrões de vulnerabilidade.

O objetivo é **aliviar, reequilibrar e expandir** a energia do usuário, de forma ética, segura e não invasiva.

Entrada Sensorial

Nem todos os dias são leves.

Por isso, o FriendApp não espera que o usuário peça ajuda: se houver consentimento, ele **acolhe preventivamente**.

Aqui, a tecnologia atua como um **campo regenerativo invisível**, oferecendo práticas, mensagens e experiências que funcionam como **cura digital vibracional**.

Estrutura dos Protocolos

Módulo	Descrição	Ativação
Mensagem Sensorial da IA Aurah	Texto acolhedor + áudio 432Hz	Score < 5 em check-in
Sugestão de Conteúdo Vibracional	Práticas guiadas (respiração, áudio, vídeo)	Score < 6 ou feedback negativo
Modo Alívio	Redução de estímulos: feed simplificado, notificações pausadas	Score < 4 persistente
Grupo "Solo Consciente"	Espaço protegido para usuários em queda vibracional	Score < 5 por mais de 7 dias
Semente de Luz	Envio de toque vibracional para amigo próximo (se opt-in ativo)	Score crítico (<3)

Fluxo Operacional

1. Usuário realiza check-in → score = 3,2 (queda).
2. IA Aurah dispara evento → regeneracao-service.
3. Service aplica matriz de decisão:
 - Se score entre 4–5 → Mensagem sensorial + conteúdo regenerativo.
 - Se score < 4 → Ativa Modo Alívio.
 - Se queda persistente → sugere Grupo Solo Consciente.
4. Logs criptografados → audit-service (auditoria ética).

Exemplos de Práticas Regenerativas

Prática	Frequência Associada	Duração	Finalidade
Respiração em 5 tempos	528Hz	3 min	Reduz ansiedade
Áudio "Flutua"	432Hz + sons de água	5 min	Alívio mental
Prática Corpo Estrela	852Hz	6 min	Elevação vibracional
Mensagem "Você é raiz e céu"	396Hz	Instantâneo	Reconexão com estabilidade
Vídeo Sol Interior	963Hz	2 min	Expansão da clareza

Pseudocódigo Simplificado

```
def ativar_regeneracao(user_id, score):
    if score < 3:
        ativar_modos_alivio(user_id)
        sugerir_grupo_solo(user_id)
    elif score < 5:
        enviar_mensagem_aurah(user_id)
        recomendar_conteudo(user_id, "regenerativo")
    elif score < 6:
        sugerir_audio(user_id, "flutua")
    registrar_evento("regeneracao_ativada", user_id, score)
```

Integrações

Sistema	Função
IA Aurah Kosmos	Detecta queda e escolhe protocolo
Feed Sensorial	Reduz estímulos ou sugere conteúdos suaves
Mapa de Frequência	Atualiza cor/energia do usuário em queda
Comunidades	Sugere Grupo Solo Consciente
Gamificação	Concede selo "Renascente" após recuperação




Exemplo Real

- João faz check-in com score 2,5 → queda crítica.
- Sistema ativa Modo Alívio: feed reduzido, notificações pausadas.
- Aurah envia mensagem:
"Sua energia pede cuidado. Respire comigo em 432Hz, estou aqui."
- João realiza prática → score sobe para 4,5 após 2 dias.
- Recebe selo simbólico **"Renascente"**.



Fechamento

 Os mecanismos de regeneração vibracional transformam quedas em oportunidades de **cura e expansão**.

Eles garantem que nenhum usuário atravessasse momentos difíceis sozinho: sempre haverá uma mão estendida, seja da IA, da comunidade ou de um mentor.



CAMADA 23 — PAINEL DE MÉTRICAS SENSORIAIS PARA ADMINISTRADORES (EXPANDIDA)



Propósito

Oferecer à equipe do FriendApp um **painel administrativo oculto** para acompanhar a saúde vibracional do ecossistema.

Ele não serve para vigiar usuários individuais, mas para monitorar **tendências coletivas**, eficácia de protocolos e **desempenho da IA Aurah Kosmos**.

É a bússola da gestão ética e operacional.



Entrada Sensorial

O painel é o **radar silencioso do campo coletivo**.

Aqui, números não são apenas métricas — são **ondas vibracionais** traduzidas em gráficos, cores e alertas, ajudando a equipe a perceber **quando o ecossistema precisa de mais luz, cuidado ou ajustes técnicos**.



Estrutura do Painel

Seção	Métricas Monitoradas	Visualização
Jornada & Expansão	Taxa de check-ins, curvas médias de frequência (7/30 dias)	Linha temporal + cores vibracionais
Regeneração & Protocolos	% eficácia de conteúdos/práticas em ≤48h	Gráfico de pulsação verde/vermelho
Comunidades & Saúde	Afinidade média, toxicidade detectada, flags da curadoria	Radar vibracional + mapa de clusters
Suporte & Ética	Chamados N1/N2/N3, SLA de Guardiões/Mentores	Tabelas e semáforos
IA Aurah Kosmos	Taxa de acerto vs. fallback humano, confiança média	Gauge de confiança
Eventos & Conexões	Participação em encontros, feedbacks pós-eventos	Barras + heatmap geográfico

Fluxo Operacional do Painel

1. Dados coletados em tempo real → telemetry pipeline.
2. Stream Processor agrega e normaliza.
3. Métricas publicadas no metrics-service.
4. Painel renderizado em Grafana/Metabase com camadas sensoriais:
 - cores vibracionais
 - animações pulsantes
5. Alertas enviados ao time admin se thresholds quebrados.

Indicadores-Chave (KPIs Vibracionais)

Indicador	Fórmula	Alvo
Índice de Expansão	Δ score médio 7d vs 30d	> +10%
Taxa de Regeneração	usuários recuperados/total em risco	> 65%
Queda Coletiva	% check-ins <3 em 24h	< 5%
Satisfação de Suporte	feedbacks positivos / total	> 85%
Confiança Aurah	respostas IA sem fallback / total	> 90%
Afinidade Comunitária	usuários ativos em clusters ≥ limiar	> 80%

Exemplo de Dashboard (JSON de Configuração)

```
{
  "panel": "Expansão Coletiva",
  "metric": "avg_score_7d",
  "thresholds": { "warn": 4.0, "crit": 3.0 },
  "visual": "line",
  "color_scheme": ["#00FF99", "#FFD700", "#FF3366"]
}
```


Integrações do Painel

Sistema	Dados Fornecidos
Checkin-Service	scores, tendências
Regeneracao-Service	protocolos ativados, eficácia
Curadoria-Service	flags de conteúdos/grupos
Suporte-Service	chamados e escalonamentos
Aurah-Gateway	inferências e confiança
Eventos-Service	participação e feedbacks

Exemplo Real

- No painel, a curva de expansão média caiu 12% em 7 dias.
- Alertas automáticos: **Queda Coletiva** ativada.
- Aurah sugere campanha regenerativa ("Semana da Respiração Solar").
- Painel mostra após 3 dias: +8% de recuperação → eficácia confirmada.

Fechamento

 O painel de métricas sensoriais é o **olho ético e estratégico** do FriendApp.

Ele garante que o time admin veja não apenas números, mas **batimentos coletivos**, tomando decisões rápidas, conscientes e vibracionais.

CAMADA 24 — CONEXÕES COM TRANSMUTAÇÃO, EVENTOS E SELOS (EXPANDIDA)

Propósito

Explicar como o sistema de **Suporte + Comunidade + Jornada Sensorial** se conecta de forma direta com os módulos de **Transmutação, Eventos e Selos**, garantindo que cada experiência de cuidado gere **expansão real, encontros significativos e reconhecimento simbólico**.

Entrada Sensorial

O suporte não é fim em si mesmo: é **portal**.

Cada queda acolhida pode virar transmutação, cada dúvida pode se tornar convite para um evento, cada superação merece ser reconhecida com um selo.

Essa camada mostra como o FriendApp **fecha o ciclo**: da dor → à cura → ao crescimento coletivo.

Estrutura de Conexões

Sistema	Como se Conecta	Exemplo de Ação
Jogo da Transmutação	Recebe triggers da IA quando padrões de bloqueio são detectados	Usuário em repetição de quedas → missão de transmutação

Sistema	Como se Conecta	Exemplo de Ação
Sistema de Eventos	Sugere encontros reais/digitais conforme estado vibracional	Usuário em solidão → convite para roda de acolhimento
Sistema de Selos	Gera conquistas simbólicas após superação ou consistência	Usuário recupera frequência após Modo Alívio → selo "Renascente"

Fluxo Operacional

1. Usuário entra em suporte e relata queda repetitiva.
2. IA Aurah → calcula Score Vulnerabilidade.
3. Se padrão repetitivo detectado:
 - Trigger → jogo-transmutacao.
 - Missão enviada: "Prática de Transmutação da Ansiedade".
4. Após conclusão da missão:
 - Timeline atualizada.
 - Selo concedido: "Guardião da Própria Luz".
 - Feed sugere evento de integração compatível.

Exemplos de Interação

Situação	Conexão Ativada
3 check-ins negativos consecutivos	Missão de Transmutação sugerida
Usuário relata solidão no suporte	Convite automático para evento de grupo
Recuperação após protocolo regenerativo	Selo simbólico concedido
Guardião apoia usuário em crise	Selo "Guardião Vibracional" atribuído

Exemplo JSON de Trigger

```
{
  "event": "user_recovered",
  "user_id": "uuid",
  "score_before": 2.5,
  "score_after": 6.1,
  "actions": [
    {"type": "selo", "id": "renascente"},
    {"type": "evento", "id": "meditacao_coletiva"}
  ]
}
```

Integrações

Sistema	Função Técnica
Transmutação	Recebe triggers de vulnerabilidade → gera missões sensoriais

Sistema	Função Técnica
Eventos	Usa dados de estado emocional → sugere experiências presenciais
Selos	Gera recompensas simbólicas → motiva consistência
Mapa de Frequência	Reflete coletivamente os avanços de cura e encontros
Feed Sensorial	Exibe conquistas e convites de forma contextualizada



Exemplo Real

- Mariana cai para score 2,8 → Modo Alívio ativado.
- Após prática, ela sobe para 5,5.
- Sistema dispara:
 - Selo **“Renascente”** concedido.
 - Evento sugerido: “Círculo da Lua Nova” (acolhimento coletivo).
 - Missão de Transmutação: “Respiração Estelar por 7 dias”.
- Linha do tempo mostra evolução → comunidade vibra junto.



Fechamento

📌 O suporte se torna ponte para experiências transformadoras.

Cada vulnerabilidade acolhida gera **transmutação**, cada passo de cura se expande em **encontros**, cada superação é celebrada em **símbolos**.

Assim, o FriendApp garante que **nenhuma jornada seja solitária**: tudo está conectado em círculos de evolução.



CAMADA 25 — MÓDULO DE FEEDBACK EMOCIONAL DO SUPORTE + CICLO DE REGENERAÇÃO FINAL (EXPANDIDA)



Propósito

Fechar o ciclo do suporte vibracional no FriendApp:

- **Coletar feedback emocional** após cada interação de suporte.
- Garantir que esse feedback ative **ajustes automáticos** na IA, nos Guardiões e nos protocolos regenerativos.
- Consolidar a jornada do usuário com **símbolos e práticas finais**, marcando cada ciclo como concluído e preparando o próximo.



Entrada Sensorial

O cuidado não termina quando a dúvida é respondida.

Ele se encerra quando o usuário **se sente visto, acolhido e fortalecido**.

Por isso, cada sessão de suporte finaliza com um momento de **feedback emocional** — um **espelho breve**, onde o usuário pode confirmar: *“Sim, agora estou melhor.”*

Esse módulo é o **último abraço digital** do ciclo de cuidado.

Estrutura do Módulo

Etapa	Elemento Sensorial	Função
Coleta	Pergunta suave: <i>“Como você se sente agora?”</i>	Captar estado pós-suporte
Registro	Emoção + escala vibracional + opção de comentário	Alimentar logs
Processamento	IA cruza resposta com atendimento dado	Avaliar eficácia
Ação	Ativar protocolo extra ou sugerir prática se necessário	Garantir evolução
Encerramento	Mensagem de gratidão + símbolo vibracional	Concluir ciclo

Fluxo Operacional

1. Usuário finaliza interação com suporte.
2. Sistema exibe painel sensorial de feedback:
 - Escala (melhor / igual / pior)
 - Emoção atual
 - Campo opcional de texto/áudio
3. Dados → feedback-service.
4. IA Aurah processa:
 - Se evolução positiva → reforço + selo simbólico.
 - Se estado negativo → protocolo regenerativo ativado.
5. Logs são criptografados → audit-service.

Dados Monitorados

Dado	Finalidade
Escala emocional pós-suporte	Avaliar eficácia do atendimento
Emoção declarada	Comparar antes/depois
Comentário opcional	Ajustar linguagem da IA
Tempo de resposta ao feedback	Detectar engajamento real
Padrão coletivo de feedback	Ajustar protocolos globais

Pseudocódigo de Avaliação

```
def avaliar_feedback(user_id, estado_antes, estado_depois):  
    if estado_depois > estado_antes:  
        conceder_selo(user_id, "Renascente")  
        return "positivo"  
    elif estado_depois == estado_antes:  
        sugerir_conteudo(user_id, "reforço")  
        return "neutro"
```



```
else:
    ativar_regeneracao(user_id)
    notificar_guardiao(user_id)
    return "negativo"
```

Integrações

Sistema	Função
IA Aurah Kosmos	Processa feedback e sugere práticas
Regeneração	Ativa protocolos adicionais se necessário
Selos	Concede recompensas simbólicas (ex.: "Resiliência")
Mapa de Frequência	Atualiza evolução final do ciclo
Audit-Service	Garante rastreabilidade ética dos feedbacks

Exemplo Real

- Usuário Mariana entra em suporte com score 3, sai com 5.
- Feedback: *"Me sinto mais calma, obrigado."*
- IA processa → evolução positiva.
- Sistema concede selo **"Renascente"**.
- Timeline mostra marco: *"Superou ciclo de queda com apoio vibracional."*

Se Mariana tivesse relatado piora, protocolo regenerativo extra seria ativado automaticamente.

Fechamento

 O módulo de feedback emocional é o **fecho dourado do ciclo de suporte**.

Ele garante que cada jornada não termine em resposta, mas em **cura, evolução e celebração simbólica**.

Aqui, o FriendApp fecha um ciclo para abrir outro, sempre mais elevado.

SEÇÃO FINAL — INTEGRAÇÕES E DEPENDÊNCIAS CÍCLICAS DO ECOSSISTEMA (EXPANDIDA)

Propósito

Mapear com **clareza absoluta** como o módulo de **Suporte + Comunidade + Jornada Sensorial** se conecta ao restante do ecossistema FriendApp, garantindo que desenvolvedores e curadores nunca se perguntem:

- *"De onde vêm esses dados?"*
- *"Para onde isso vai?"*
- *"Quem consome essa informação?"*

Entrada Sensorial

O FriendApp é um **organismo vivo**.

Cada sistema é um **órgão pulsante**, e a saúde de um depende do outro.

As integrações são o **sangue energético** que corre pelas veias digitais do ecossistema, mantendo tudo em harmonia.

Dependências (inputs necessários)

Sistema Dependido	Dados Recebidos	Uso no Suporte + Jornada
Cadastro Consciente	Nome, idade, consentimentos, aura inicial	Ativar painel de privacidade e perfil vibracional
Teste Energético	Frequência dominante, arquétipo vibracional	Basear recomendações iniciais
Mapa de Frequência	Clusters coletivos, estado global	Harmonizar comunidades e feed
Consent-Service	Opt-ins de privacidade	Validar se IA pode agir proativamente
Aurah Kosmos	Inferências emocionais, insights	Motor central do suporte vibracional

Saídas (dados enviados para outros módulos)

Sistema Destinatário	Dados Enviados	Finalidade
Feed Sensorial	Recomendações, insights, alertas positivos	Ajustar conteúdos exibidos
Gamificação & Selos	Eventos de superação, consistência	Premiar evolução vibracional
Jogo da Transmutação	Triggers de bloqueios repetitivos	Sugerir missões de transmutação
Sistema de Eventos	Estado vibracional + preferências	Convidar para encontros compatíveis
Comunidades	Afinidade energética	Sugerir grupos afins ou de apoio
Painel Admin Ético	Estatísticas agregadas	Monitorar saúde coletiva, não individual

Mecanismos Técnicos de Integração

- **Eventos Assíncronos (Event Bus)**
 - `event.checkin.created` → Timeline, Recomendações, Mapa, Selos
 - `event.support.opened` → Guardiões/Mentores, Audit
 - `event.regeneracao.started` → Timeline, Metrics
- **APIs REST/gRPC**
 - `/support/feedback` → envia resultados de interações
 - `/recommendations` → requisita grupos, conteúdos, eventos
 - `/consent/validate` → verifica permissões antes de qualquer ação
- **Padrões de Segurança**
 - Consent Token obrigatório em cada requisição.
 - JWT assinado para autenticação entre microserviços.
 - Logs criptografados (AES-256).

Ciclos de Entrada e Saída

flowchart TD

A[Check-in Vibracional] → B(Aurah Kosmos)

B → C[Feed Sensorial]

B → D[Mapa de Frequência]

B → E[Recomendações de Comunidades]

E → F[Eventos & Encontros]

B → G[Gamificação & Selos]

G → H[Jogo da Transmutação]

H → B

D → I[Painel Admin Ético]

- Cada ação do usuário gera **múltiplos efeitos cíclicos**.
- O resultado de um módulo retroalimenta outro (ex.: Transmutação → IA Aurah → Feed).

Regras Éticas do Ciclo

1. Nenhum sistema age sem **consentimento validado**.
2. Todos os dados são processados com **propósito declarado**.
3. As dependências nunca são usadas para vigilância, apenas para **cura e expansão**.
4. Estatísticas individuais são anonimizadas em relatórios administrativos.

Fechamento Final

 Esta seção é o **manual das conexões invisíveis**: a prova de que o FriendApp é um ecossistema **cíclico, ético e integrado**.

Nada aqui funciona isolado. Tudo vibra em conjunto.