

# MANUAL TÉCNICO DEFINITIVO — SISTEMA DE SEGURANÇA VIBRACIONAL (FRIENDAPP)

## CAMADA 01 — DESCRIÇÃO GERAL + PROPÓSITO

### Introdução

O **Sistema de Segurança Vibracional** do FriendApp foi concebido para ser a **espinha dorsal da proteção energética, emocional e informacional** de todo o ecossistema. Diferente dos modelos tradicionais de segurança digital, ele não é punitivo, nem centrado em vigilância, mas atua como um **campo inteligente de blindagem e prevenção**, que combina **IA Vibracional, lógica de risco matemática, escudos dinâmicos e insights transparentes ao usuário**.

Ele deve operar em tempo real, 24/7, garantindo que:

- Cada interação esteja protegida contra riscos técnicos e energéticos.
- Perfis falsos ou incoerentes não comprometam a experiência do coletivo.
- Menores de idade tenham uma proteção reforçada, mas sem segregação.
- O usuário se sinta **amparado e no controle**, nunca vigiado ou punido.

### Objetivos do Sistema

1. **Proteção Total:** atuar sobre riscos técnicos (fraude, bots) e vibracionais (colapsos, incoerências).
2. **Prevenção Proativa:** usar IA preditiva para antecipar cenários de risco e agir antes do problema se materializar.
3. **Educação Acolhedora:** transformar incidentes em oportunidades de autoconhecimento, com mensagens explicativas.
4. **Transparência:** oferecer ao usuário clareza sobre os ajustes feitos pelo sistema através do **Painel de Confiança Vibracional**.
5. **Escalabilidade:** sustentar milhares de usuários simultâneos sem comprometer latência (< 300ms).





### Escopo de Atuação

O sistema cobre todos os pontos críticos do FriendApp:

- **Cadastro Consciente:** verificação de idade e documentos, ativação de proteções invisíveis.
- **Feed Sensorial:** filtragem e curadoria em tempo real, adaptada à frequência do usuário.
- **Chat e Mensagens Vibracionais:** moderação semântica e vibracional de conteúdos.
- **Sistema de Conexões Autênticas:** validação energética de compatibilidade.
- **Eventos e Grupos Bora:** monitoramento coletivo, prevenção de colapsos em massa.
- **Aurah Kosmos (IA central):** núcleo de leitura vibracional e tomada de decisão do sistema.

### Arquitetura Conceitual

O sistema é construído sobre **4 pilares principais**:

| Pilar   | Função  |
|---|---|
|  IA de Prevenção Vibracional | Detecta padrões incoerentes e calcula scores de risco em tempo real.      |
|  Escudos Dinâmicos           | Ativados automaticamente em resposta a eventos detectados.                |
|  Painel de Confiança         | Transparência ao usuário: mostra insights, níveis de confiança e alertas. |
|  Observabilidade             | Coleta e analisa logs técnicos + vibracionais para evolução contínua.     |

## Diferenciais Técnicos

1. **Matriz de Risco Vibracional** — modelo matemático que atribui peso a cada sinal do usuário (tempo de resposta, linguagem, hesitação, intenção, etc.), gerando o `score_risco_inicial`.
2. **Proteção Invisível + Transparência** — os escudos atuam silenciosamente, mas os efeitos são comunicados no Painel do usuário para evitar frustração.
3. **Integração com APIs Externas** — verificação de identidade (DUC/DCO) feita via parceiros como Unico ou Idwall, reduzindo riscos de vazamento.
4. **Logs Estruturados** — todos os eventos são registrados em formato JSON padronizado para auditoria, suporte e melhoria do sistema.

## Integrações Obrigatórias

- **IA Aurah Kosmos** → valida as leituras vibracionais e aciona escudos.
- **Feed Sensorial** → recebe filtros e curadorias em tempo real.
- **Mapa de Frequência** → atualizado conforme oscilações e eventos coletivos.
- **Segurança Técnica (firewalls)** → protege contra automações e ataques externos.
- **Observabilidade (Grafana + Prometheus)** → dashboards internos de monitoramento.

## Fechamento da Camada

O **Sistema de Segurança Vibracional** é mais que uma camada de proteção: é um **guardião invisível, técnico e humano ao mesmo tempo**, capaz de transformar segurança em confiança.

Ele une lógica técnica, matemática de risco e cuidado energético, sendo a base para que o FriendApp se mantenha **seguro, autêntico e humano em escala global**.

# CAMADA 02 — ARQUITETURA CONCEITUAL DA SEGURANÇA






## Introdução

A **Arquitetura Conceitual do Sistema de Segurança Vibracional** descreve como os módulos técnicos e energéticos se organizam para proteger o FriendApp em todas as suas dimensões: **cadastro, feed, conexões, chat e eventos**.

Diferente de modelos convencionais, ela é pensada para ser **modular, escalável e observável**, permitindo expansão futura sem perda de coerência.

## Pilares Estruturais

A arquitetura foi projetada em **4 pilares técnicos + 1 pilar de experiência do usuário**:

| Pilar   | Função Técnica  | Benefício ao Usuário                  |
|---|---|---------------------------------------|
|  IA de Prevenção Vibracional | Lê padrões de comportamento + linguagem + energia em tempo real | Atua como guardião invisível          |
|  Escudos Dinâmicos           | Aplicam restrições técnicas e vibracionais graduais             | Protege sem censurar                  |
|  Painel de Confiança         | Dá visibilidade ao usuário das ações tomadas pela IA            | Transparência e autonomia             |
|  Observabilidade             | Monitora e loga todas as decisões do sistema                    | Permite auditoria e evolução contínua |
|  UX Sensorial Adaptativa     | Ajusta sons, cores e efeitos da UI conforme o estado energético | Usuário sente cuidado, não punição    |

## Camadas Técnicas da Arquitetura

### ◆ Frontend (Camada de Experiência)

- **Framework:** Flutter + FriendFX
- **Função:** adaptar a UI em tempo real conforme frequência do usuário.
- **Exemplo:** ao detectar oscilação vibracional, cores mudam para tons azuis suaves.

### ◆ Backend (Camada de Processamento)

- **Tecnologias:** Node.js + Go (microserviços)
- **Função:** motor de risco, validação de documentos, escudos dinâmicos.
- **Exemplo:** cálculo do `score_risco_inicial` e ativação de firewall vibracional.

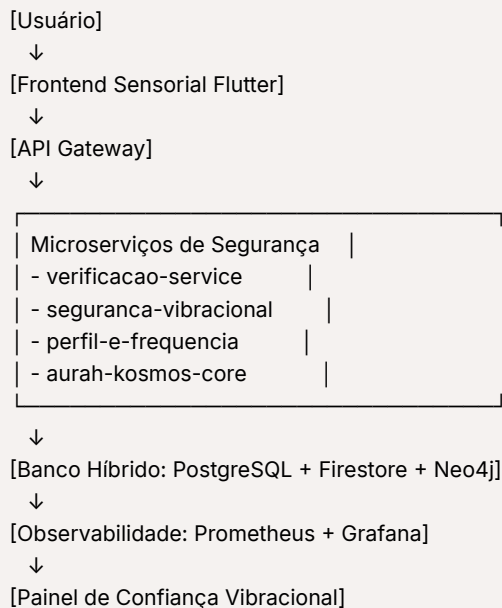
### ◆ Banco de Dados Híbrido

- **PostgreSQL:** dados estruturados de cadastro e verificações
- **Firestore (NoSQL):** estados vibracionais em tempo real
- **Neo4j (grafo):** conexões energéticas entre usuários

### ◆ Orquestração

- **Kubernetes + Istio:** escalabilidade automática e roteamento por microserviço.
- **EventBus + Kafka:** garante que logs e eventos de risco sejam distribuídos em tempo real.

## Fluxo Arquitetural Macro



## Integrações Essenciais

- **IA Aurah Kosmos** → valida padrões vibracionais, ativa escudos.
- **Feed Sensorial** → recebe filtros e curadoria em tempo real.
- **Mapa de Frequência** → atualizado conforme interações.
- **Sistema de Conexões** → bloqueia perfis incoerentes automaticamente.
- **APIs Externas de Verificação** → valida documentos e identidade (ex.: Unico, IdWall).

## Segurança por Design

- **Criptografia:** TLS 1.3 em trânsito, AES-256 em repouso.
- **Firewalls Vibracionais:** bloqueiam tentativas de manipulação energética.
- **Isolamento por Cluster:** usuários incoerentes são isolados em grafos paralelos.
- **Logs Criptografados:** todos os eventos registrados em JSON padronizado.

## Fechamento da Camada

A Arquitetura Conceitual garante que o Sistema de Segurança Vibracional seja **robusto, modular e humano ao mesmo tempo**.

Combinando **IA, escudos dinâmicos, bancos híbridos e observabilidade**, o sistema consegue proteger milhares de usuários simultâneos sem perder a **sensação de cuidado e autenticidade** que é a essência do FriendApp.

# CAMADA 03 — BLINDAGEM VIBRACIONAL: FILOSOFIA E LÓGICA

## Introdução

A **Blindagem Vibracional** é o coração do sistema de proteção do FriendApp.

Ela não é uma barreira punitiva ou um bloqueio rígido, mas sim um **mecanismo adaptativo e sensível**, capaz de:

- **Prevenir abusos** (comportamentais, linguísticos, energéticos).
- **Atuar em diferentes níveis de intensidade** conforme o risco detectado.
- **Oferecer feedback transparente ao usuário**, evitando frustração.
- **Manter a coerência do ecossistema coletivo**, sem afetar a liberdade individual.

## Filosofia da Blindagem

A filosofia por trás desse módulo é inspirada em três princípios:

1. **Proteção antes da reação** → A blindagem detecta sinais de risco antes que a experiência seja comprometida.
2. **Acolhimento em vez de punição** → O usuário nunca sente que está sendo censurado, mas sim cuidado.
3. **Flexibilidade progressiva** → A intensidade da blindagem se adapta ao nível de risco (leve, moderado, grave).

## Tipos de Blindagem Ativa

| Nível    | Condição Detectada  | Ação Executada no Sistema   |
|----------|---|---|
| Leve     | Oscilações emocionais pequenas, hesitação, linguagem neutra | UI suavizada (tons frios), redução de estímulos, sons binaurais leves   |
| Moderada | Uso de palavras negativas, tempo de resposta anômalo        | Feed entra em modo silencioso + delay de conexões novas                 |
| Grave    | Linguagem abusiva, colapso vibracional, tentativa de fraude | Chat temporariamente bloqueado, feed recolhido, escudos ativos visíveis |

## Lógica Técnica de Ativação

### Matriz de Condições

A ativação da blindagem segue uma **fórmula baseada no** `score_risco_inicial` (definida na Camada 06), cruzada com sinais contextuais:

```

if score_risco_inicial < 0.4:
    ativar("blindagem_leve")
elif score_risco_inicial >= 0.4 and score_risco_inicial < 0.75:
    ativar("blindagem_moderada")
else:
    ativar("blindagem_grave")

```

## Variáveis Monitoradas

- **tempo\_resposta** → anomalias (<1s ou >30s em perguntas simples).
- **linguagem\_negativa** → frequência de palavras agressivas ou densas.
- **inconsistencia\_escolhas** → divergência entre intenção declarada e comportamento real.
- **verificado\_duc** → peso de redução de risco para usuários autenticados.

## Resposta Sensorial na UI

Cada blindagem gera **respostas perceptíveis**, mas sutis, para o usuário:

| Blindagem | Cor Primária | Som Ativado       | Feedback Aurah Kosmos                                       |
|-----------|--------------|-------------------|---|
| Leve      | Azul claro   | 528Hz (cura)      | "Seu campo está oscilando, vamos suavizar sua experiência." |
| Moderada  | Lilás/azul   | 432Hz (proteção)  | "Percebemos instabilidade, sua timeline ficará mais calma." |
| Grave     | Azul escuro  | 396Hz (blindagem) | "Sua segurança está em prioridade, ajustamos conexões."     |

## Integrações

- **IA Aurah Kosmos** → valida risco, escolhe tipo de blindagem.
- **Feed Sensorial** → ativa ou reduz conteúdos conforme nível de risco.
- **Chat e Conexões** → bloqueio parcial ou delays de interação.
- **Logs de Observabilidade** → registra tipo de blindagem, duração e resultados.

## Exemplo de Log JSON

```

{
  "usuario_id": "8d12-ff29",
  "blindagem_ativada": "moderada",
  "score_risco_inicial": 0.58,
  "variaveis": {
    "tempo_resposta": 1.2,
    "linguagem_negativa": true,
    "inconsistencia_escolhas": false,
    "verificado_duc": false},
  "timestamp": "2025-09-02T10:34:22Z"
}

```

## Fechamento da Camada

A Blindagem Vibracional é o **primeiro nível de defesa do FriendApp**.

Sua lógica é **matemática, preditiva e transparente**, equilibrando segurança com experiência.

Ela é o mecanismo que garante que **nenhum usuário seja exposto a abusos ou riscos energéticos**, sem que isso seja sentido como censura ou limitação de liberdade.

## ◆ CAMADA 04 — DIAGRAMA TÉCNICO (VISUAL E DESCRITO)

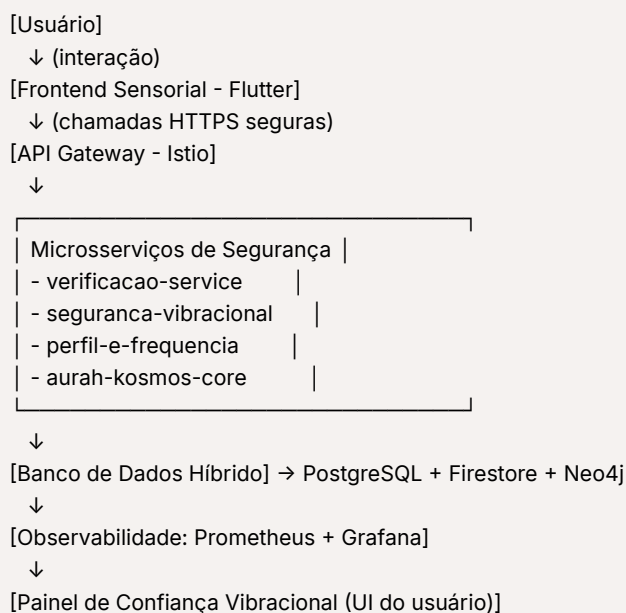
### Introdução

O **Diagrama Técnico** traduz em uma visão visual e lógica como o Sistema de Segurança Vibracional funciona na prática.

Ele mostra a comunicação entre **usuário, frontend, backend, microserviços, bancos de dados e IA Aurah Kosmos**, além do fluxo de ativação dos **escudos vibracionais**.

Essa camada é essencial para que engenheiros, arquitetos e auditores compreendam **onde cada decisão é tomada, como os dados circulam e onde ficam registrados**.

### Fluxo Operacional Completo



### Descrição dos Componentes

#### ◆ Usuário

- Interage pelo app (toques, textos, voz, tempo de resposta).
- Dados coletados: idade, linguagem, comportamento, intenção.

#### ◆ Frontend Sensorial (Flutter + FriendFX)

- Interface adaptativa que responde ao estado energético.
- Capta padrões de interação (ex: velocidade de scroll, toques agressivos).

#### ◆ API Gateway (Istio)

- Responsável pelo roteamento de chamadas entre frontend e microserviços.
- Faz **rate limiting** para evitar flood de requests.

#### ◆ Microserviços de Segurança

- **verificacao-service** → valida DUC/DCO via API externa (Unico, IdWall).

- **seguranca-vibracional-service** → aplica lógica da matriz de risco, decide blindagem.
- **perfil-e-frequencia-service** → gera o perfil vibracional inicial.
- **aurah-kosmos-core** → núcleo de IA que interpreta sinais energéticos.

#### ◆ Banco de Dados Híbrido

- **PostgreSQL** → dados estruturados (cadastro, idade, verificações).
- **Firestore** → estados vibracionais em tempo real.
- **Neo4j** → grafos de conexões autênticas.

#### ◆ Observabilidade

- **Prometheus** coleta métricas de performance.
- **Grafana** exibe dashboards para auditar fluxos e gatilhos de segurança.

#### ◆ Painel de Confiança Vibracional

- UI do usuário onde aparecem insights, alertas e indicadores de confiabilidade.
- Exemplo: "Aurah percebeu que sua energia está mais sensível, ajustamos seu feed."

### Exemplo de Evento Técnico

1. Usuário posta conteúdo com linguagem incoerente.
2. Frontend envia payload ao **API Gateway**.
3. **seguranca-vibracional-service** processa texto + sinais vibracionais.
4. Aurah Kosmos gera score de risco ( **score\_risco\_inicial = 0.82** ).
5. Blindagem **grave** é ativada.
6. Firestore registra **blindagem\_ativa = grave**.
7. Log enviado ao Grafana → aparece em dashboard interno.
8. Painel do usuário exibe insight transparente.

### Exemplo de Log JSON

```
{
  "usuario_id": "91b4-cc11",
  "evento": "post_incoerente",
  "score_risco_inicial": 0.82,
  "acao": "ativar_blindagem_grave",
  "detalhes": {
    "palavras_negativas": true,
    "tempo_resposta": 0.9,
    "verificado_duc": false
  },
  "painel_usuario": "Feed ajustado para reduzir conteúdo intenso.",
  "timestamp": "2025-09-02T14:26:53Z"
}
```

### Fechamento da Camada

O **Diagrama Técnico** garante uma visão clara do ecossistema de segurança, mostrando como as informações fluem, onde são processadas e como são devolvidas em forma de proteção transparente para o usuário.

Com essa visualização, qualquer dev ou arquiteto pode compreender rapidamente a lógica, os pontos de decisão e os registros envolvidos.

## CAMADA 05 — FLUXO SENSORIAL DA SEGURANÇA NO APP

### Introdução

O **Fluxo Sensorial da Segurança** é a forma como o sistema traduz ações técnicas de proteção em **experiências perceptíveis na interface do usuário**.

A ideia é que a pessoa **sinta a presença da segurança**, mas nunca como censura: a UI se adapta, os sons mudam, as interações ficam mais suaves, e a IA explica o que está acontecendo em linguagem acessível.

### Objetivo

Garantir que a ativação dos escudos vibracionais seja acompanhada de **sinais sensoriais coerentes**, que mantenham o usuário em equilíbrio e transmitam segurança.

### Estados e Respostas Sensoriais

| Estado Vibracional Detectado   | Ação Sensorial no App  | Objetivo   |
|--|--|--|
| <b>Oscilação Leve (score &lt; 0.4)</b>                               | UI suaviza cores (azul claro), animações ficam mais lentas, som 528Hz    | Reduz estímulos e cria sensação de calma         |
| <b>Risco Moderado (<math>0.4 \leq \text{score} &lt; 0.75</math>)</b> | Feed entra em "modo silencioso", sons 432Hz, escudo translúcido discreto | Evitar sobrecarga emocional e reforçar contenção |
| <b>Crise Grave (score <math>\geq 0.75</math>)</b>                    | Tela assume tons escuros, efeitos reduzidos, som 396Hz, bloqueio parcial | Ativar proteção máxima sem parecer punição       |
| <b>Recuperação</b>   | UI retorna gradualmente ao normal, sons mais expansivos (639Hz)          | Dar sensação de retomada de equilíbrio           |

### Fluxo Operacional Sensorial

```
[Leitura Vibracional]
↓
[Classificação de Estado]
↓
[Definição do Escudo Técnico]
↓
[Tradução Sensorial na UI] → Cor, Som, Animação, Feedback IA
↓
[Log Registrado + Painel do Usuário Atualizado]
```

### Exemplo Técnico

#### Evento: Oscilação Moderada

- Score de risco = 0.62
- IA Aurah Kosmos recomenda **blindagem moderada**.
- Feed entra em **modo silencioso** → posts novos não aparecem por 10 min.
- UI muda para tons lilás/azul frio.
- Som de fundo em 432Hz é ativado.
- Painel do usuário exibe:



"Aurah percebeu que sua energia está instável. Ajustamos o ritmo da sua experiência para manter seu campo protegido."

## Estrutura de Dados

```
{
  "usuario_id": "f9d1-3412",
  "estado_vibracional": "moderado",
  "score": 0.62,
  "acoes_sensoriais": {
    "ui_cor": "#4F5D75",
    "som": "432Hz",
    "feed_status": "modo_silencioso",
    "animacoes": "lentas"
  },
  "painel_feedback": "Energia instável, feed ajustado por 10min",
  "timestamp": "2025-09-02T18:11:54Z"
}
```

## Integrações Ativas

- **Aurah Kosmos** → interpreta o estado e escolhe resposta sensorial.
- **Feed Sensorial** → ativa modo silencioso ou expande conteúdos leves.
- **Chat Vibracional** → ajusta sugestões de conexão conforme estado.
- **Painel de Confiança Vibracional** → mostra feedback em tempo real.

## Fechamento da Camada

O **Fluxo Sensorial de Segurança** garante que a proteção seja percebida como **cuidado acolhedor**.

Cada ajuste de cor, som ou animação funciona como **signal sensorial**, alinhando a experiência emocional do usuário ao estado vibracional detectado, sem punição e sem bloqueios arbitrários.

# CAMADA 06 — MOTOR DE DECISÃO INICIAL (ENERGIA + PERFIL + VERIFICAÇÃO)

## Introdução

O **Motor de Decisão Inicial** é o núcleo que interpreta todas as informações coletadas durante o cadastro e onboarding do usuário.

Ele transforma dados brutos (idade, respostas emocionais, tempo de reação, documentos, teste de personalidade) em **variáveis técnicas** que definem:

- O **nível de proteção inicial** (escudos vibracionais ativos).
- O **grau de confiança** do perfil.
- A **trilha vibracional de entrada** no app.
- As primeiras permissões em **feed, chat, conexões e eventos**.

Esse motor deve operar em **tempo inferior a 400ms**, garantindo que o usuário não perceba latência entre sua entrada e a ativação da experiência.

## Entradas do Motor

| Variável                | Origem Técnica                             | Uso no Motor                                |
|-------------------------|--|---|
| idade                   | Cadastro (data de nascimento)              | Ativa fluxo adolescente ou adulto           |
| verificado_duc          | Verificação externa (API Unico/IdWall)     | Reduz peso de risco, aumenta confiabilidade |
| tempo_resposta          | Logs do cadastro e onboarding              | Detecta bots ou usuários incoerentes        |
| linguagem_emocional     | Análise semântica (NLP)                    | Classifica intenção vibracional             |
| inconsistencia_escolhas | Teste de Personalidade x respostas rápidas | Marca dissonância de intenção               |
| frequencia_inicial      | Resultado do teste vibracional             | Base do perfil energético                   |
| zona_horaria            | Localização x horário de login             | Detecta incoerências de geolocalização      |

## Fórmula do Score de Risco Inicial

A partir das variáveis, é calculado o **score\_risco\_inicial** (0 a 1), que orienta a blindagem:

$$\text{score\_risco\_inicial} = (0.25 \times \text{tempo\_resposta\_anormalo}) + (0.25 \times \text{linguagem\_negativa}) + (0.20 \times \text{inconsistencia\_escolhas}) + (0.20 \times \text{fator\_geolocalizacao}) - (0.30 \times \text{verificado\_duc})$$
$$\text{score\_risco\_inicial} = (0.25 \times \text{tempo\_resposta\_anormalo}) + (0.25 \times \text{linguagem\_negativa}) + (0.20 \times \text{inconsistencia\_escolhas}) + (0.20 \times \text{fator\_geolocalizacao}) - (0.30 \times \text{verificado\_duc})$$

- **tempo\_resposta\_anormalo**: 0 se normal, até 1 se respostas muito rápidas ou lentas.
- **linguagem\_negativa**: 0 a 1 conforme análise semântica (NLP).
- **inconsistencia\_escolhas**: 0 a 1 conforme divergência entre teste e onboarding.
- **fator\_geolocalizacao**: 0 a 1 se detectar login incoerente com região/horário.
- **verificado\_duc**: reduz risco (-0.3).

## Classificação do Score

| Score de Risco                 | Nível de Blindagem Ativada | Permissões Iniciais                              |
|--------------------------------|----------------------------|--|
| < 0.4                          | Blindagem Leve             | Feed completo, conexões abertas                  |
| $0.4 \leq \text{score} < 0.75$ | Blindagem Moderada         | Feed silencioso, conexões limitadas              |
| $\geq 0.75$                    | Blindagem Grave            | Feed reduzido, chat limitado, eventos bloqueados |

## Exemplo Prático

- Usuário declara 35 anos.
- Escolhe intenção “amizades leves”.
- Em respostas abertas usa termos agressivos.
- Tempo de resposta médio = 0.7s.
- Documento **não verificado**.

Resultado:

$$\text{score\_risco\_inicial} = 0.25(0.8) + 0.25(0.7) + 0.20(0.6) + 0.20(0.3) - 0.30(0) = 0.61$$
$$\text{score\_risco\_inicial} = 0.25(0.8) + 0.25(0.7) + 0.20(0.6) + 0.20(0.3) - 0.30(0) = 0.61$$

Classificação: **Risco Moderado** → blindagem moderada ativada.

## Exemplo de Log JSON

```
{
  "usuario_id": "c12f-88e9",
  "score_risco_inicial": 0.61,
  "variaveis": {
    "tempo_resposta_anomalo": 0.8,
    "linguagem_negativa": 0.7,
    "inconsistencia_escolhas": 0.6,
    "fator_geolocalizacao": 0.3,
    "verificado_duc": false},
  "blindagem_ativada": "moderada",
  "timestamp": "2025-09-02T20:15:07Z"
}
```

## Integrações Ativas

- **Aurah Kosmos** → interpreta score e ajusta feedback.
- **Feed Sensorial** → reduz postagens intensas em blindagem moderada.
- **Chat** → atraso de mensagens em blindagem grave.
- **Mapa de Frequência** → registra vibração inicial no grafo.
- **Painel de Confiança Vibracional** → exhibe insights e nível de confiabilidade.

## Fechamento da Camada

O **Motor de Decisão Inicial** transforma dados brutos em um **perfil técnico-vibracional confiável**, ativando proteções adequadas e configurando a experiência inicial.

Com fórmulas claras e logs estruturados, elimina subjetividade e garante que devs possam implementar sem dúvidas.

# CAMADA 07 — ESTRUTURA DAS PERGUNTAS VIBRACIONAIS RÁPIDAS (PRÉ-TESTE)

## Introdução

As **Perguntas Vibracionais Rápidas** funcionam como uma etapa intermediária entre o cadastro consciente e o **Teste de Personalidade Energética completo**.

O objetivo é **capturar o estado emocional imediato** do usuário e alinhar suas intenções antes do teste mais profundo, sem gerar fricção ou sensação de burocracia.

Essa etapa é crítica porque:

- Alimenta o **Motor de Decisão Inicial** (Camada 06).
- Ajuda a calibrar a intensidade e a linha de estímulos do Teste Energético.
- Permite ativar **escudos preliminares** se padrões incoerentes forem detectados.

## Objetivos Técnicos das Perguntas

1. Mapear **intenção inicial** de uso (amizades, expansão, resenha, etc.).
2. Medir o **estado vibracional atual** (oscilações emocionais).
3. Identificar **dissonâncias imediatas** entre fala e escolha.
4. Alimentar variáveis para cálculo do `score_risco_inicial`.

## Estrutura das Perguntas

| Categoria                    | Formato Técnico                 | Uso no Sistema   |
|------------------------------|---------------------------------|--|
| Intenção de Entrada          | Múltipla escolha com ícones     | Define propósito principal → conecta com trilhas de feed   |
| Estado Emocional Atual       | Escala deslizante (0 a 10)      | Alinha frequência do usuário → ajusta estímulos do teste   |
| Estilo de Experiência        | Ícones com toque longo          | Capta nível de abertura (sensorial, direto, introspectivo) |
| Grau de Interação Esperado   | Múltipla escolha                | Marca perfil social (introvertido, explorador, observador) |
| Frequência com Redes Sociais | Alternativas + comentário livre | Mede influência externa → ajusta peso no score de risco    |

## Fórmulas de Peso

As respostas são convertidas em **scores normalizados (0 a 1)**.

Cada categoria alimenta variáveis no motor de decisão:

$score\_presenca\_emocional = (0.4 \times estado\_emocional) + (0.3 \times consistencia\_intencao) + (0.2 \times estilo\_experiencia) + (0.1 \times grau\_interacao)$   
 $score\_presenca\_emocional = (0.4 \times estado\_emocional) + (0.3 \times consistencia\_intencao) + (0.2 \times estilo\_experiencia) + (0.1 \times grau\_interacao)$

$score\_presenca\_emocional = (0.4 \times estado\_emocional) + (0.3 \times consistencia\_intencao) + (0.2 \times estilo\_experiencia) + (0.1 \times grau\_interacao)$

- **estado\_emocional**: normalizado de 0 (muito baixo) a 1 (muito alto).
- **consistencia\_intencao**: 1 se intenção declarada condiz com respostas abertas; 0 se dissonante.
- **estilo\_experiencia**: valores atribuídos conforme escolha (sensorial = 0.8, direto = 0.5, introspectivo = 0.6).
- **grau\_interacao**: 0.3 introvertido, 0.7 explorador, 0.5 observador.

## Exemplo de Pergunta e Registro

**Pergunta:** "Como você gostaria de se sentir aqui dentro?"

- Opções: "leve", "conectado", "seguro", "inspirado"
- Usuário escolhe: "**conectado**"

**Registro no Firestore:**

```
{
  "usuario_id": "3f21-881a",
  "pergunta": "intencao_entrada",
  "resposta": "conectado",
  "score": 0.85,
  "timestamp": "2025-09-02T20:54:12Z"
}
```

## Integrações Ativas

- **Aurah Kosmos** → interpreta padrões de resposta e tempo de escolha.
- **Motor de Decisão Inicial (Camada 06)** → usa scores para calcular risco.
- **Teste de Personalidade** → recebe como input a intenção inicial e adapta estímulos.
- **Painel de Confiança Vibracional** → poderá exibir insights sobre intenção declarada.

## Logs de Observabilidade

Todos os dados das perguntas são registrados em logs técnicos para auditoria:

```
{
  "usuario_id": "3f21-881a",
  "respostas_vibracionais": {
    "intencao": "conectado",
    "estado_emocional": 7,
    "estilo_experiencia": "sensorial",
    "grau_interacao": "explorador",
    "uso_redes_sociais": "baixo"
  },
  "score_presenca_emocional": 0.72,
  "timestamp": "2025-09-02T20:56:33Z"
}
```

## ✓ Fechamento da Camada

As **Perguntas Vibracionais Rápidas** são o **primeiro checkpoint energético real do usuário**.

Elas alimentam a matriz de risco, ajustam o teste subsequente e garantem que a jornada comece **alinhada, protegida e personalizada**.

## ◆ CAMADA 08 — GATILHOS DE IA E ESCUDOS ATIVADOS PELAS RESPOSTAS VIBRACIONAIS

### 🕒 Introdução

Após a coleta das **Perguntas Vibracionais Rápidas** (Camada 07), o sistema aciona uma camada de **inteligência automática** que interpreta as respostas e ativa **escudos invisíveis ou visíveis** conforme o risco identificado.

Essa etapa é fundamental porque garante que **mesmo antes do teste completo** já exista proteção ativa, evitando exposição precoce a conteúdos incoerentes ou abusivos.

### 🔬 Tipos de Gatilhos da IA

| Nome do Gatilho       | Condição Técnica Detectada                           | Ação do Sistema   |
|-----------------------|--|---|
| trigger_social        | Resposta emocional baixa + intenção de isolamento    | Ativa <b>escudo social</b> → reduz exposição a conteúdos de massa     |
| trigger_explorador    | Alta abertura + perfil não verificado                | Ativa <b>escudo explorador</b> → bloqueia conexões incoerentes        |
| trigger_dissonancia   | Inconsistência entre intenção e linguagem usada      | IA recalibra feed e alerta painel com insight vibracional             |
| trigger_nebuloso      | Tempo de resposta irregular + hesitação frequente    | Ativa <b>escudo nebuloso</b> → UI desacelera, sons de 432Hz aplicados |
| trigger_autenticidade | Usuário não verificou DUC e mostra padrões suspeitos | Reforço de firewall invisível + recomendação de verificação           |

### 🎲 Fórmula de Ativação dos Escudos

$$\text{escudo\_final} = \text{argmax}\{\text{peso\_social}, \text{peso\_explorador}, \text{peso\_dissonancia}, \text{peso\_nebuloso}, \text{peso\_autenticidade}\}$$

$$\text{escudo\_final} = \text{argmax}\{\text{peso\_social}, \text{peso\_explorador}, \text{peso\_dissonancia}, \text{peso\_nebuloso}, \text{peso\_autenticidade}\}$$

$$\text{escudo\_final} = \text{argmax}\{\text{peso\_social}, \text{peso\_explorador}, \text{peso\_dissonancia}, \text{peso\_nebuloso}, \text{peso\_autenticidade}\}$$

Onde cada peso é calculado a partir de variáveis normalizadas:

- **peso\_social** =  $0.3 \times (1 - \text{estado\_emocional}) + 0.2 \times \text{isolamento\_declarado}$
- **peso\_explorador** =  $0.4 \times \text{abertura} + 0.3 \times (1 - \text{verificado\_duc})$


- **peso\_dissonancia** =  $0.5 \times \text{inconsist\^encia\_escolhas} + 0.3 \times \text{linguagem\_negativa}$
- **peso\_nebuloso** =  $0.4 \times \text{hesita\~ao} + 0.3 \times \text{tempo\_resposta\_anormalo}$
- **peso\_autenticidade** =  $0.6 \times (1 - \text{verificado\_duc}) + 0.2 \times \text{inconsist\^encia\_escolhas}$

## Exemplo de Execução

- Estado emocional = 3/10
- Intenção = "explorar conexões novas"
- Usuário **sem verificação DUC**
- Tempo de resposta médio: 1.1s (levemente anômalo)

Resultado:

- **peso\_social** = 0.41
- **peso\_explorador** = 0.58
- **peso\_dissonancia** = 0.25
- **peso\_nebuloso** = 0.21
- **peso\_autenticidade** = 0.47

 Escudo ativado = **Explorador**

## Exemplo de Log JSON

```
{
  "usuario_id": "1a2b-44c9",
  "gatilho": "trigger_explorador",
  "score_risco_inicial": 0.59,
  "escudo_ativado": "explorador",
  "variaveis": {
    "estado_emocional": 0.3,
    "abertura": 0.7,
    "verificado_duc": false,
    "tempo_resposta_anormalo": 0.2,
    "linguagem_negativa": 0.1
  },
  "timestamp": "2025-09-02T21:12:18Z"
}
```

## Integrações Ativas

- **Aurah Kosmos** → analisa respostas, atribui pesos e escolhe escudo.
- **Feed Sensorial** → reduz intensidade de conteúdos incoerentes.
- **Chat Vibracional** → pode aplicar delay em conexões suspeitas.
- **Painel de Confiança Vibracional** → exibe insight transparente ao usuário:

“Ativamos um campo explorador para proteger sua energia. Você pode revisar isso em suas configurações.”

## Fechamento da Camada

Os **Gatilhos de IA e Escudos Ativos** representam a primeira camada real de defesa do FriendApp.

Eles permitem proteção imediata, sem depender de verificações posteriores, garantindo que desde a entrada no app o usuário esteja **seguro, acolhido e consciente** das proteções aplicadas.

## ◆ CAMADA 09 — INTEGRAÇÃO COM O TESTE DE PERSONALIDADE ENERGÉTICA

### Introdução

O **Teste de Personalidade Energética** é o módulo mais profundo de entrada do FriendApp.

A **integração com o Sistema de Segurança Vibracional** garante que:

- A experiência do teste seja **adaptada ao estado atual do usuário** (leve, moderado ou crítico).
- Os **escudos ativados na Camada 08** influenciem na escolha de estímulos visuais, auditivos e linguísticos.
- Todos os dados coletados sejam usados tanto para gerar o **Mapa de Frequência** (Camada 10) quanto para atualizar os **scores de risco**.

### Fluxo de Integração

```
[Respostas Rápidas] → [Motor de Decisão Inicial] → [Escudos Ativos]
↓
Início do Teste Energético (/api/personality/start)
↓
Aurah Kosmos ajusta estímulos conforme escudos ativos
↓
Usuário responde (toque, tempo, hesitação)
↓
Resultados gravados no Firestore + PostgreSQL
↓
Cálculo do perfil vibracional + atualização do score de risco
```

### Endpoints de API

| Endpoint                     | Método | Descrição  |
|------------------------------|--------|--|
| /api/personality/start       | POST   | Cria sessão do teste, define trilha sensorial inicial    |
| /api/personality/inputs      | POST   | Recebe idade, escudos ativos, intenção e contexto        |
| /api/personality/session/:id | GET    | Retorna status atual do teste                            |
| /api/personality/results     | POST   | Envia resultados para cálculo de perfil + score de risco |

### Variáveis de Entrada para o Teste

| Variável           | Origem                        | Uso no Teste                                 |
|--------------------|-------------------------------|--|
| idade              | Cadastro                      | Define limites de estímulos                  |
| verificado_duc     | API de terceiros              | Reduz risco e amplia acesso                  |
| escudos_ativos     | Camada 08                     | Ajusta cores, sons e tipos de estímulos      |
| intencao_entrada   | Perguntas rápidas (Camada 07) | Define foco da linha energética              |
| frequencia_inicial | Motor de Decisão (Camada 06)  | Ponto de referência para comparar resultados |

### Personalização Sensorial

Dependendo dos escudos ativos, o teste é adaptado:

- **Escudo Social** → reduz intensidade de imagens, usa cores suaves.

- **Escudo Explorador** → aumenta elementos de foco e clareza, reduz ambiguidade.
- **Escudo Dissonância** → IA introduz perguntas de checagem cruzada.
- **Escudo Nebuloso** → tempo de resposta permitido é maior, sons 432Hz aplicados.

## Exemplo de Ajuste Preditivo

- Usuário entrou com `escudo_dissonancia`.
- IA detecta incoerência entre intenção declarada e linguagem.
- O teste ativa **perguntas extras de validação** para confirmar perfil.

Resultado:

- Se consistência for confirmada, risco reduzido em 0.2.
- Se incoerência se repetir, risco aumenta em 0.3.

## Exemplo de Log JSON

```
{
  "usuario_id": "a91d-2c71",
  "teste_sessao": "sess-88291",
  "escudos_ativos": ["dissonancia"],
  "ajustes_aplicados": {
    "extra_perguntas": true,
    "tempo_resposta_max": 15,
    "som": "432Hz"
  },
  "resultado": {
    "perfil_vibracional": "Explorador Empático",
    "frequencia": 6.8,
    "consistencia": 0.72
  },
  "timestamp": "2025-09-02T21:45:12Z"
}
```

## Integrações

- **Aurah Kosmos** → define trilha sensorial adaptada.
- **Feed Sensorial** → recebe perfil vibracional após conclusão.
- **Mapa de Frequência** → inicia geração de ponto energético.
- **Painel de Confiança Vibracional** → pode exibir mensagem:

“Concluímos sua leitura vibracional inicial. Sua energia principal foi registrada como ‘Explorador Empático’.”

## Fechamento da Camada

A integração entre o **Teste de Personalidade Energética** e o **Sistema de Segurança Vibracional** garante que o processo seja **imersivo, protegido e matematicamente consistente**.

As respostas alimentam diretamente o motor de risco e a IA Aurah Kosmos, gerando uma experiência **segura e personalizada desde o início**.



# CAMADA 10 — GERAÇÃO DO MAPA DE FREQUÊNCIA INICIAL (PÓS-TESTE)

## Introdução

O **Mapa de Frequência Inicial** é o artefato central gerado após o Teste de Personalidade Energética.

Ele traduz as respostas do usuário em **valores técnicos quantificáveis** que representam seu estado vibracional no momento da entrada no FriendApp.

Esse mapa é usado como base para:

- Configurar o **Feed Sensorial**.
- Definir as primeiras **conexões autênticas**.
- Atualizar o **grafo global vibracional**.
- Alimentar a **IA Aurah Kosmos** com dados para curadoria personalizada.
- Ativar **escudos de segurança personalizados** conforme estabilidade energética.

## Estrutura Técnica do Mapa

| Elemento                          | Tipo         | Fonte de Dados                           | Finalidade Técnica                                     |
|-----------------------------------|--------------|--|--|
| <code>frequencia_numerica</code>  | Float (0–10) | Resultado do teste + tempo de resposta   | Valor central da energia                               |
| <code>perfil_vibracional</code>   | Enum         | Teste de Personalidade Energética        | Define arquétipo (Ex.: Empata, Guardião, Visionário)   |
| <code>campo_de_ressonancia</code> | Array        | Respostas emocionais e palavras-chave    | Lista padrões dominantes de energia emocional          |
| <code>grau_coerencia</code>       | Float (0–1)  | IA Aurah Kosmos                          | Mede consistência entre intenção, linguagem e escolhas |
| <code>expansao_social</code>      | Int (0–100)  | Teste + estilo de interação              | Nível de abertura para conexões                        |
| <code>zona_geo_energetica</code>  | String       | Localização + frequência média da região | Posiciona o usuário em clusters coletivos              |
| <code>hash_mapa</code>            | Hash String  | Função SHA-256                           | Garante integridade e anonimização do perfil           |

## Fórmulas Matemáticas

### Frequência Numérica

$$\text{frequencia\_numerica} = (0.4 \times \text{media\_respostas}) + (0.3 \times \text{consistencia}) + (0.2 \times \text{tempo\_resposta}) + (0.1 \times \text{estabilidade\_emocional})$$
$$\text{frequencia\_numerica} = (0.4 \times \text{media\_respostas}) + (0.3 \times \text{consistencia}) + (0.2 \times \text{tempo\_resposta}) + (0.1 \times \text{estabilidade\_emocional})$$

- **media\_respostas** → normalização das escolhas do teste (0–10).
- **consistencia** → correlação entre intenção inicial e linguagem usada.
- **tempo\_resposta** → penalização de valores anômalos (<1s ou >20s).
- **estabilidade\_emocional** → análise semântica de palavras positivas/negativas.

### Grau de Coerência

$$\text{grau\_coerencia} = 1 - \frac{(\text{inconsistencias\_detectadas})}{(\text{total\_perguntas})}$$
$$\text{grau\_coerencia} = 1 - \frac{(\text{inconsistencias\_detectadas})}{(\text{total\_perguntas})}$$

- Varia entre 0 (totalmente incoerente) e 1 (altamente coerente).

## Exemplo de Saída (Firestore)

```
{
  "usuario_id": "56a3-4412",
  "frequencia_numerica": 7.42,
  "perfil_vibracional": "Empata",
  "campo_de_ressonancia": ["afetividade", "introspecção", "escuta"],
  "grau_coerencia": 0.88,
  "expansao_social": 63,
  "zona_geo_energetica": "São Paulo | Sul Vibracional",
  "hash_mapa": "ab3efc23a9c9a0deff..."
}
```

## Fluxo Operacional

```
[Resultado do Teste]
↓
[Motor de Cálculo Vibracional]
↓
[IA Aurah Kosmos valida consistência]
↓
[Mapa de Frequência Inicial Gerado]
↓
[Salvo em Firestore + PostgreSQL + Neo4j]
↓
[Atualiza Feed, Conexões e Painel do Usuário]
```

## Integrações

- **Aurah Kosmos** → valida coerência e interpreta resultados.
- **Feed Sensorial** → recebe frequência e perfis compatíveis.
- **Mapa Global (Neo4j)** → posiciona usuário em clusters coletivos.
- **Segurança Vibracional** → ativa escudos baseados em estabilidade.
- **Painel de Confiança** → exibe resumo amigável ao usuário.

Exemplo de insight mostrado no painel:

“Sua energia inicial foi registrada como 7.4, com perfil Empata. Ajustamos seu feed para conexões mais suaves.”

## Exemplo de Log JSON

```
{
  "evento": "mapa_gerado",
  "usuario_id": "56a3-4412",
  "frequencia_numerica": 7.42,
  "perfil_vibracional": "Empata",
  "grau_coerencia": 0.88,
  "expansao_social": 63,
  "timestamp": "2025-09-02T22:05:41Z"
}
```

## ✓ Fechamento da Camada

A **Geração do Mapa de Frequência Inicial** é o passo que transforma o usuário em um **nó vibracional ativo dentro do ecossistema**.

É aqui que segurança, personalização e transparência se unem, entregando ao mesmo tempo proteção, insights e coerência técnica.

## ◆ CAMADA 11 — ENTRADA VIBRACIONAL NO ECOSSISTEMA DO APP

### 📍 Introdução

A **Entrada Vibracional no Ecossistema** é o momento em que o usuário deixa de estar em fase de cadastro/teste e passa a ser **um nó ativo** dentro do FriendApp.

Aqui, o sistema conecta o **Mapa de Frequência Inicial** (Camada 10) com os módulos práticos: feed, conexões, chat e mapa global.

É nesta camada que:

- O usuário **ganha posição energética no grafo coletivo**.
- Os escudos ativados anteriormente são **ajustados para operação contínua**.
- A IA Aurah Kosmos começa a **monitorar e sugerir trilhas em tempo real**.

### 🔄 Fluxo Operacional

[Mapa de Frequência Inicial Gerado]  
↓  
[Motor de Entrada Vibracional]  
↓  
1. Atualiza Grafo Global (Neo4j)  
2. Configura Feed Sensorial Inicial  
3. Ativa Conexões Autênticas sugeridas  
4. Sincroniza com IA Aurah Kosmos  
5. Registra no Painel de Confiança Vibracional

### 📊 Níveis de Abertura Inicial

A entrada não é igual para todos: depende do score vibracional, grau de coerência e verificação de identidade.

| Nível de Entrada | Condição Técnica  | Ações Ativas   |
|------------------|---|--|
| <b>Protegido</b> | $\text{score\_risco\_inicial} \geq 0.75$ OU sem verificação DUC | Feed filtrado, conexões limitadas, escudos reforçados        |
| <b>Neutro</b>    | $0.4 \leq \text{score\_risco\_inicial} < 0.75$                  | Feed moderado, conexões progressivas, escudos dinâmicos      |
| <b>Expansivo</b> | $\text{score\_risco\_inicial} < 0.4$ + DUC verificado           | Feed completo, conexões amplas, trilhas vibracionais abertas |

### 🧠 Interação com Aurah Kosmos

A IA passa a operar em **modo contínuo**, aplicando:

- Monitoramento em ciclos de 30s a 1min.
- Recomendações energéticas adaptadas (conteúdos, conexões).
- Feedback transparente via Painel de Confiança.

Exemplo de insight:

"Sua energia está em 71, estável. Abrimos conexões compatíveis em São Paulo."

## Integrações Ativas

- **Feed Sensorial** → recebe configurações iniciais.
- **Conexões Autênticas** → sugere até 3 conexões iniciais coerentes.
- **Mapa Global (Neo4j)** → adiciona usuário ao cluster energético.
- **Painel de Confiança** → mostra status inicial de entrada.
- **Logs/Observabilidade** → registra evento `entrada_vibracional`.

## Exemplo de Log JSON

```
{
  "usuario_id": "9c2a-b812",
  "evento": "entrada_vibracional",
  "score_risco_inicial": 0.38,
  "nivel_entrada": "expansivo",
  "acoes": {
    "feed_status": "completo",
    "conexoes_iniciais": ["u884", "u199", "u221"],
    "mapa_global": "cluster_sul_vibracional"
  },
  "painel_feedback": "Energia estável, conexões abertas",
  "timestamp": "2025-09-02T22:32:51Z"
}
```

## Fechamento da Camada

A **Entrada Vibracional** é o rito de passagem do usuário: de iniciante para **participante ativo do campo coletivo**.

Ela une segurança, personalização e posicionamento no grafo, garantindo que cada pessoa entre no FriendApp com uma experiência **segura, contextualizada e transparente**.

# CAMADA 12 — TRIGGERS OCULTOS DE SEGURANÇA E INCOERÊNCIA INICIAL

## Introdução

Após a **Entrada Vibracional** (Camada 11), o sistema ativa um conjunto de **triggers ocultos** que monitoram os primeiros dias do usuário dentro do FriendApp.

Essa etapa é crítica porque muitos comportamentos incoerentes ou abusivos só se revelam **após a entrada no ecossistema**, quando o usuário interage com feed, chat e grupos.

Os triggers funcionam como **sensores invisíveis** que analisam linguagem, frequência, comportamento e contexto, aplicando medidas automáticas de contenção sempre que detectam incoerências.

## Objetivos da Camada

1. Detectar **inconsistências entre cadastro e comportamento real**.
2. Prevenir **abuso precoce** em feed e chat.
3. Garantir **que menores de idade estejam protegidos**, mesmo se omitirem idade.

4. Acionar **escudos automáticos** sem quebrar a experiência do usuário.

## Triggers Ativos

| Nome do Trigger                      | Condição de Ativação   | Ação Automática Aplicada  |
|--------------------------------------|--|---|
| <code>trigger_dissonancia</code>     | Intenção declarada ≠ comportamento no feed/chat                | Modulação do feed, blindagem moderada e log de incoerência      |
| <code>trigger_sombra</code>          | Linguagem agressiva, sexual ou manipulativa em chat/posts      | Mensagem bloqueada + filtro invisível + alerta IA               |
| <code>trigger_fraude</code>          | Geolocalização oscilante + comportamento automático (bot)      | Conta entra em observação, feed reduzido e conexões suspensas   |
| <code>trigger_excesso</code>         | Volume anormal de interações em pouco tempo                    | Suspende sugestões de conexão + feed silencioso por 15 min      |
| <code>trigger_coerencia_idade</code> | Usuário diz ser adulto, mas padrões indicam adolescência (<18) | Ativa escudo juvenil invisível + feed filtrado + alerta interno |

## Fórmulas de Detecção

### Exemplo — Trigger de Excesso

$\text{trigger\_excesso} = \text{interacoes} / \text{minutos} > 5$

$\text{trigger\_excesso} = \text{minutos} / \text{interacoes} > 5$

Se o usuário fizer **mais de 5 interações por minuto em 10 min seguidos**, o trigger é acionado.

### Exemplo — Trigger de Dissonância

$\text{dissonancia} = \text{intencao\_declarada} \neq \text{comportamento\_observado}$   
 $\text{dissonancia} = \text{intencao\_declarada} \neq \text{comportamento\_observado}$

$\text{dissonancia} = \text{intencao\_declarada} \neq \text{comportamento\_observado}$

- Intenção = "amizade leve"
- Comportamento = envio de mensagens sexuais/agressivas

Resultado: `dissonancia = true` → **risco +0.3 no score**.

## Exemplo de Log JSON

```
{
  "usuario_id": "bb29-77d1",
  "trigger_ativado": "trigger_sombra",
  "acao_tomada": "mensagem_bloqueada",
  "score_risco_inicial": 0.78,
  "detalhes": {
    "conteudo": "Você é um lixo",
    "tipo": "mensagem_chat",
    "classificacao": "linguagem_agressiva"
  },
  "painel_feedback": "Bloqueamos uma mensagem com linguagem incoerente.",
  "timestamp": "2025-09-02T23:05:41Z"
}
```

## Integrações

- **Aurah Kosmos** → interpreta padrões incoerentes, aumenta score de risco.
- **Feed Sensorial** → modulado automaticamente após triggers de risco.

- **Chat Vibracional** → aplica delay ou bloqueia mensagens nocivas.
- **Painel de Confiança** → mostra insights amigáveis:

“Detectamos excesso de interações. Ajustamos o ritmo para preservar sua energia.”

## ✓ Fechamento da Camada

Os **Triggers Ocultos de Segurança** são responsáveis por manter o FriendApp protegido **nas primeiras interações reais**.

Eles garantem que incoerências, abusos ou fraudes sejam tratados em tempo real, com medidas automáticas, registros técnicos e feedback transparente ao usuário.

# ◆ CAMADA 13 — CONEXÃO COM A JORNADA SENSORIAL DE CRESCIMENTO

## 🧭 Introdução

Após a entrada vibracional (Camada 11) e a ativação de triggers de segurança (Camada 12), o FriendApp conecta o usuário à **Jornada Sensorial de Crescimento**.

Essa jornada é um **conjunto progressivo de microações, conteúdos e desafios leves**, desenhados para fortalecer a estabilidade energética do usuário e reduzir riscos de reincidência em incoerências.

## 🎯 Objetivos da Camada

1. **Oferecer caminhos de evolução energética** para usuários em blindagem moderada ou grave.
2. **Reforçar a segurança invisível** por meio de interações educativas.
3. **Dar transparência** ao processo, transformando filtros em insights claros.
4. **Engajar o usuário** em práticas de autodescoberta e reequilíbrio.

## 🔄 Fluxo Operacional

```

[Entrada Vibracional]
↓
[Verificação dos Triggers Ativos]
↓
[Aurah Kosmos define trilha de crescimento]
↓
[Jornada Sensorial personalizada é iniciada]
↓
[Usuário realiza microações → IA recalibra perfil]
↓
[Mapa de Frequência e Feed atualizados]
  
```

## 🧩 Estrutura da Jornada

| Etapa da Jornada        | Conteúdo/Atividade                                | Efeito Técnico  |
|-------------------------|---|---|
| Dia 1 — Boas-vindas     | Mensagem personalizada da Aurah Kosmos            | Reduz score de risco em 0.1 se interação for positiva |
| Dia 3 — Reflexão Guiada | Exercício de autoexpressão (texto ou áudio curto) | Detecta consistência vibracional → recalibra feed     |

| Etapa da Jornada         | Conteúdo/Atividade  | Efeito Técnico   |
|--------------------------|---|--|
| Dia 5 — Microação Social | Curtir/comentar em post compatível                        | Estimula expansão saudável, aumenta score de confiança |
| Dia 7 — Feedback IA      | Insight no Painel: "Sua energia cresceu 12% esta semana." | Mostra evolução ao usuário, engaja para continuidade   |

## Fórmula de Evolução da Jornada

Cada ação concluída reduz o risco vibracional:

$$\text{score\_risco\_ajustado} = \text{score\_risco\_inicial} - (0.05 \times \text{acoes\_concluidas}) + (0.02 \times \text{incoerencias\_novas})$$

$$= \text{score\_risco\_inicial} - (0.05 \times \text{acoes\_concluidas}) + (0.02 \times \text{incoerencias\_novas})$$

$$\text{score\_risco\_ajustado} = \text{score\_risco\_inicial} - (0.05 \times \text{acoes\_concluidas}) + (0.02 \times \text{incoerencias\_novas})$$

- **ações\_concluidas** → cada microação realizada reduz risco.
- **incoerencias\_novas** → penalização se novos triggers forem ativados.

## Exemplo de Log JSON

```
{
  "usuario_id": "d91f-55a7",
  "jornada_id": "journey-019",
  "trilha": "expansao_social",
  "acoes": [
    {"dia": 1, "acao": "mensagem_boas_vindas", "concluida": true},
    {"dia": 3, "acao": "reflexao_guiada", "concluida": false}
  ],
  "score_risco_ajustado": 0.52,
  "feedback_painel": "Sua energia está mais estável, continue nessa trilha.",
  "timestamp": "2025-09-02T23:34:07Z"
}
```

## Integrações

- **Aurah Kosmos** → define conteúdo e tom da jornada.
- **Feed Sensorial** → libera gradualmente conteúdos mais intensos conforme evolução.
- **Mapa de Frequência** → ajusta posição do usuário no cluster conforme score.
- **Painel de Confiança** → mostra evolução percentual ("+8% estabilidade").

## Fechamento da Camada

A **Jornada Sensorial de Crescimento** transforma a segurança em **processo educativo e engajador**.

O usuário sente que está evoluindo, não sendo punido.

Isso garante **retenção, transparência e maturidade vibracional**, fortalecendo a confiança no ecossistema FriendApp.

# CAMADA 14 — SISTEMA DE ONBOARDING SENSORIAL (EXPERIÊNCIA DE ENTRADA)

## Introdução

O **Onboarding Sensorial** é a primeira experiência prática do usuário no FriendApp após o cadastro e geração do Mapa de Frequência Inicial.

Aqui, a segurança vibracional se conecta diretamente com a **interface de boas-vindas**, transformando o ingresso no ecossistema em uma experiência:

- **Imersiva** → UI adaptada à frequência inicial.
- **Segura** → escudos invisíveis já ativos e calibrados.
- **Transparente** → Painel de Confiança mostra de forma amigável como o app cuida do usuário.
- **Engajadora** → sugere microações leves para integração social imediata.

## Objetivos da Camada

1. Guiar o usuário de forma **sensorial e educativa** na sua primeira sessão.
2. **Explicar visualmente** como sua energia foi registrada e como ela será usada.
3. **Ativar o feed e conexões iniciais** com base na sua frequência.
4. Garantir que ele entre no FriendApp já **confiando no campo protetor**.

## Estrutura do Onboarding

O onboarding é dividido em **5 etapas principais**, exibidas em telas interativas (scroll ou swipe):

| Etapa                       | Conteúdo / Atividade                                   | Ação Técnica no Backend  |
|-----------------------------|--|--|
| 1. Boas-vindas Vibracionais | Mensagem da Aurah Kosmos com nome + frequência inicial | Loga evento <code>onboarding_iniciado</code>                       |
| 2. Explicação do Mapa       | Visualização do valor vibracional e perfil arquétipo   | Carrega dados do Firestore (frequência, coerência)                 |
| 3. Proteção Ativada         | Feedback amigável sobre escudos invisíveis ativos      | Consulta <code>seguranca-vibracional-service</code>                |
| 4. Recursos Disponíveis     | Demonstração interativa: Feed, Conexões, Bora, Eventos | Pré-carrega dados de APIs correspondentes                          |
| 5. Primeira Ação            | Sugestão prática: seguir alguém ou interagir no feed   | Loga microação inicial no <code>perfil-e-frequencia-service</code> |

## Personalização Sensorial

O onboarding é adaptado conforme o perfil vibracional:

| Perfil Vibracional | Cores da UI           | Som Base         | Linguagem IA                       |
|--------------------|-----------------------|------------------|------------------------------------|
| Empata             | Tons lilás e verdes   | 528Hz (cura)     | "Você é uma energia acolhedora"    |
| Guardião           | Tons azuis e dourados | 432Hz (proteção) | "Você traz segurança para o campo" |
| Visionário         | Tons roxos e cianos   | 639Hz (expansão) | "Sua energia inspira novas ideias" |

## Integrações

- **Aurah Kosmos** → gera mensagens personalizadas por perfil.
- **Feed Sensorial** → pré-carregado com conteúdos compatíveis.
- **Mapa de Frequência** → usado para definir cores e sons da UI.
- **Segurança Vibracional** → mostra no painel que a proteção já está ativa.
- **Painel de Confiança** → exibe indicadores como "Confiabilidade Alta / Média / Em Ajuste".

## Exemplo de Log JSON

```
{  
  "usuario_id": "f27b-889a",
```



```
"evento": "onboarding_finalizado",
"frequencia_inicial": 7.1,
"perfil_vibracional": "Visionário",
"escudos_ativos": ["explorador"],
"primeira_acao": "curtir_post_inicial",
"painel_feedback": "Proteção vibracional ativada. Feed ajustado.",
"timestamp": "2025-09-02T23:55:41Z"
}
```

## ✓ Fechamento da Camada

O **Onboarding Sensorial** é o elo entre a segurança invisível e a experiência consciente do usuário.

Ele traduz escudos, scores e frequências em **feedback visual, sonoro e textual**, garantindo que a pessoa entre no FriendApp **segura, confiante e engajada**.

## ◆ CAMADA 15 — FEED INICIAL E CURADORIA SENSORIAL PÓS-CADASTRO

### 🧭 Introdução

O **Feed Inicial** é a primeira experiência social do usuário dentro do FriendApp.

Ele precisa equilibrar **acolhimento, segurança e personalização**, evitando que novos membros sejam expostos a conteúdos pesados ou desconexos logo no início.

Aqui, a **Segurança Vibracional** atua junto à **IA Aurah Kosmos** para entregar uma curadoria inicial baseada no **Mapa de Frequência** (Camada 10) e nos **Escudos Ativos** (Camada 08).

### 🎯 Objetivos da Camada

1. Construir um **feed inicial seguro** para novos usuários.
2. Exibir apenas **conteúdos compatíveis com a frequência de entrada**.
3. Usar o feed como ferramenta de **educação vibracional invisível**.
4. Reduzir chances de abandono precoce, garantindo uma experiência leve.

### 🔄 Fluxo Operacional

```
[Mapa de Frequência Inicial]
↓
[Motor de Curadoria Vibracional]
↓
1. Filtra conteúdos incoerentes
2. Seleciona posts compatíveis com frequência
3. Insere mensagens educativas da IA Aurah Kosmos
4. Sugere microações sociais leves
↓
[Feed Inicial exibido ao usuário]
```

## 🧩 Fontes de Personalização

| Fonte de Dados                 | Peso (%) | Função no Feed                                       |
|--------------------------------|----------|--|
| Mapa de Frequência Inicial     | 30%      | Define ressonância energética dos conteúdos exibidos |
| Perfil Vibracional (arquétipo) | 25%      | Ajusta linguagem e tom dos posts                     |
| Perguntas Vibracionais Rápidas | 20%      | Garante alinhamento com intenção declarada           |
| Estado Emocional Atual         | 15%      | Ajusta intensidade dos conteúdos                     |
| Escudos Ativos                 | 10%      | Oculto conteúdos incoerentes, abusivos ou excessivos |

## Fórmula do Score de Compatibilidade de Conteúdo

$$\text{compatibilidade\_conteudo} = (0.4 \times \text{proximidade\_frequencia}) + (0.3 \times \text{alinhamento\_perfil}) + (0.2 \times \text{intencao\_entrada}) - (0.2 \times \text{risco\_conteudo})$$

$$\text{compatibilidade\_conteudo} = (0.4 \times \text{proximidade\_frequencia}) + (0.3 \times \text{alinhamento\_perfil}) + (0.2 \times \text{intencao\_entrada}) - (0.2 \times \text{risco\_conteudo})$$

$$\text{compatibilidade\_conteudo} = (0.4 \times \text{proximidade\_frequencia}) + (0.3 \times \text{alinhamento\_perfil}) + (0.2 \times \text{intencao\_entrada}) - (0.2 \times \text{risco\_conteudo})$$

- **proximidade\_frequencia:** diferença entre frequência do post e do usuário.
- **alinhamento\_perfil:** compatibilidade entre arquétipo do usuário e tipo de conteúdo.
- **intencao\_entrada:** coerência entre intenção declarada e tema do post.
- **risco\_conteudo:** peso negativo para conteúdos intensos (violência, densidade emocional).

Conteúdos só entram no feed se `compatibilidade_conteudo ≥ 0.6`.

## Tipos de Conteúdo Exibidos

| Tipo de Conteúdo        | Critério de Exibição   |
|-------------------------|--|
| Mensagem do Sistema     | Sempre presente: "Seu campo está protegido e alinhado."          |
| Posts Compatíveis       | Usuários com frequência semelhante ( $\pm 0.5$ )                 |
| Pílulas Educativas      | Conteúdo leve gerado pela IA Aurah (ex.: frases + vídeos curtos) |
| Sugestões de Microações | Curtir/comentar em posts compatíveis, entrar em grupo leve       |
| Feedbacks do Painel     | Insights suaves: "Sua energia está em 71, ótima para conexões."  |

## Exemplo de Log JSON

```
{
  "usuario_id": "a82f-9910",
  "evento": "feed_inicial_gerado",
  "conteudos_exibidos": [
    {"id_post": "p101", "compatibilidade": 0.82, "tipo": "post_compativel"},
    {"id_post": "p055", "compatibilidade": 0.76, "tipo": "pilula_educativa"},
    {"id_post": "s001", "compatibilidade": 1.0, "tipo": "mensagem_sistema"}
  ],
  "escudos_ativos": ["explorador"],
  "painel_feedback": "Feed inicial gerado com foco em leveza e segurança.",
  "timestamp": "2025-09-02T23:59:11Z"
}
```

## Integrações

- **Aurah Kosmos** → gera mensagens e pílulas educativas personalizadas.
- **Feed Sensorial Service** → organiza os conteúdos e aplica filtros.
- **Mapa de Frequência** → usado para calcular proximidade entre usuários.

- **Painel de Confiança** → mostra transparência: "Filtramos 2 posts incoerentes para proteger seu campo."

## ✓ Fechamento da Camada

O **Feed Inicial** é o primeiro espelho social do usuário no FriendApp.

Ele precisa transmitir **segurança, leveza e compatibilidade energética**, evitando ruídos que poderiam gerar abandono precoce.

Com fórmulas claras, logs técnicos e integração com a IA Aurah, o sistema garante que a primeira impressão seja sempre **segura, acolhedora e personalizada**.

## ◆ CAMADA 16 — CRIAÇÃO DO MAPA DE FREQUÊNCIA INICIAL E SINCRONIZAÇÃO COM O GRAFO

### 📍 Introdução

O **Mapa de Frequência Inicial** é transformado, nesta etapa, em um **nó energético ativo** dentro do grafo global do FriendApp.

Essa sincronização garante que o usuário não seja apenas um perfil isolado, mas parte de uma **rede viva e dinâmica**, onde sua vibração influencia e é influenciada por outras.

### 🎯 Objetivos da Camada

1. Criar e registrar o **nó individual** do usuário no grafo global.
2. Sincronizar dados vibracionais com clusters de frequência similares.
3. Ativar escudos invisíveis em conexões incoerentes.
4. Garantir **consistência entre bancos híbridos** (PostgreSQL, Firestore e Neo4j).

### 🔄 Fluxo Operacional

```
[Mapa de Frequência Inicial Gerado]
↓
[Motor de Sincronização]
↓
1. Salva dados no Firestore e PostgreSQL
2. Cria nó no Neo4j (grafo global)
3. Busca clusters de proximidade vibracional
4. Define conexões iniciais (arestas)
5. Ativa escudos em vínculos incoerentes
6. Atualiza Painel de Confiança do Usuário
```

## 🧩 Estrutura Técnica

### ◆ Banco de Dados

**Firestore — Documento** `usuarios_estado_vibracional`

```
{
  "usuario_id": "u1292",
  "frequencia": 6.9,
  "perfil_vibracional": "Explorador",
  "grau_coerencia": 0.84,
```

```
"cluster_inicial": "sul_vibracional",
"hash_mapa": "4f91...ee"
}
```

## Neo4j — Nó e Arestas

```
CREATE (:Usuario {
  id: "u1292",
  frequencia: 6.9,
  perfil: "Explorador",
  coerencia: 0.84
})

MATCH (u:Usuario {id:"u1292"}), (c:Cluster {nome:"sul_vibracional"})
CREATE (u)-[:PERTENCE]->(c)
```

### ◆ Algoritmo de Clusterização

O usuário é posicionado em clusters por proximidade:

$$\text{cluster\_id} = \arg \min (| \text{freq\_usuario} - \text{freq\_cluster} | )$$
$$\text{cluster\_id} = \arg \min (| \text{freq\_usuario} - \text{freq\_cluster} | )$$

- **freq\_usuario**: frequência numérica do usuário.
- **freq\_cluster**: frequência média do cluster existente.

Se a diferença for  $> 1.5$ , cria-se um **novo cluster** automaticamente.

### ◆ Escudos em Conexões Incoerentes

Antes de validar conexões, o sistema aplica filtro:

```
for conexao in conexoes_sugeridas:
  if abs(freq_usuario - freq_conexao) > 2.0:
    bloquear_conexao(conexao)
```

## Exemplo de Log JSON

```
{
  "usuario_id": "u1292",
  "evento": "sincronizacao_grafo",
  "cluster_alocado": "sul_vibracional",
  "conexoes_criadas": ["u201", "u455", "u802"],
  "conexoes_bloqueadas": ["u188"],
  "score_risco": 0.42,
  "timestamp": "2025-09-03T00:12:33Z"
}
```

## Integrações

- **Aurah Kosmos** → confirma coerência vibracional antes da alocação.
- **Feed Sensorial** → ajustado de acordo com cluster coletivo.

- **Conexões Autênticas** → inicia sugestões a partir das arestas criadas.
- **Painel de Confiança** → exibe:

“Você foi conectado ao cluster Sul Vibracional, com 3 conexões iniciais sugeridas.”

## ✓ Fechamento da Camada

A **Sincronização com o Grafo** garante que o usuário seja imediatamente inserido em um ecossistema energético coerente.

Esse processo equilibra **matemática de clusterização, escudos de proteção e transparência via painel**, criando o elo entre a jornada individual e o coletivo vibracional do FriendApp.

# ◆ CAMADA 17 — INTEGRAÇÃO COM A IA AURAH KOSMOS PARA LEITURA INICIAL E ATIVAÇÃO VIBRACIONAL

## 🧭 Introdução

A **IA Aurah Kosmos** é o núcleo inteligente do FriendApp.

Nesta camada, ela recebe os dados do **Mapa de Frequência Inicial** (Camada 10) e da **Sincronização com o Grafo** (Camada 16), ativando a **primeira leitura vibracional contínua** do usuário.

A integração garante que Aurah Kosmos:

- Converta perfis energéticos em **assinaturas únicas**.
- Ative trilhas de crescimento e proteção personalizadas.
- Se conecte a todos os módulos (feed, chat, conexões, eventos).
- Monitore oscilações e acione escudos em tempo real.

## 🎯 Objetivos da Camada

1. **Gerar a Assinatura Vibracional Única (hash energético).**
2. Ativar a **trilha inicial personalizada** para cada usuário.
3. Sincronizar a leitura vibracional com feed, chat e grafo.
4. Registrar logs para auditoria e análise de evolução.

## 🔄 Fluxo Operacional

```

[Mapa de Frequência Inicial]
  ↓
[aurah-kosmos-core]
  ↓
1. Gera assinatura vibracional (hash)
2. Define trilha inicial
3. Ativa escudos compatíveis
4. Sincroniza com feed e conexões
5. Envia feedback ao Painel do Usuário
  
```

## 🧩 Estrutura Técnica

### ◆ Assinatura Vibracional

A assinatura é um hash energético calculado a partir de múltiplas variáveis:

assinatura=SHA256(frequencia+perfil\_vibracional+campo\_ressonancia+grau\_coerencia+timestamp)assinatura = SHA256(frequencia + perfil\_vibracional + campo\_ressonancia + grau\_coerencia + timestamp)

assinatura=SHA256(frequencia+perfil\_vibracional+campo\_ressonancia+grau\_coerencia+timestamp)

Exemplo de saída:

ae1f28df94b2c19e7c73a3d8c91f21e2e...

## ◆ Trilha Inicial

Aurah define **até 2 trilhas vibracionais** de entrada, ex.:

- Cura Emocional
- Expansão Social
- Expressão Criativa
- Foco e Estabilidade

Cada trilha influencia o conteúdo exibido no feed e sugestões de conexões.

## 📡 APIs Envolvidas

| Endpoint                      | Método | Função                                      |
|-------------------------------|--------|---|
| /api/aurah/ativar_leitura     | POST   | Inicia análise vibracional completa         |
| /api/aurah/gerar_assinatura   | POST   | Cria assinatura energética única            |
| /api/aurah/trilha/definir     | POST   | Define trilha inicial do usuário            |
| /api/feed/vibracional/aplicar | POST   | Ajusta feed de acordo com assinatura/trilha |
| /api/conexoes/preparar        | POST   | Sugere conexões iniciais coerentes          |

## 🇧🇷 Exemplo de Log JSON

```
{
  "usuario_id": "a91d-2c71",
  "evento": "aurah_ativada",
  "assinatura_vibracional": "ae1f28df94b2...",
  "perfil": "Explorador Empático",
  "trilhas": ["cura_emocional", "expressao_criativa"],
  "escudos_ativos": ["moderado"],
  "feed_status": "ajustado",
  "conexoes_sugeridas": ["u292", "u401", "u578"],
  "timestamp": "2025-09-03T00:25:42Z"
}
```

## 📡 Integrações

- **Feed Sensorial** → recebe assinatura e trilhas para personalizar conteúdos.
- **Conexões Autênticas** → inicia sugestões com base na assinatura.
- **Mapa Global (Neo4j)** → vincula assinatura ao cluster vibracional.
- **Painel de Confiança** → exibe feedback amigável:

“Sua assinatura vibracional foi ativada. Estamos cuidando para que suas conexões reflitam sua energia.”

## ✅ Fechamento da Camada

A **integração com a IA Aurah Kosmos** garante que a segurança vibracional seja **inteligente, personalizada e contínua** desde o primeiro login.

A assinatura vibracional e as trilhas definidas permitem que o usuário navegue no FriendApp de forma **única, coerente e protegida**, enquanto todos os módulos são atualizados em tempo real.

## ◆ CAMADA 18 — SISTEMA DE CADASTRO CONSCIENTE + PROTEÇÃO VIBRACIONAL INVISÍVEL PARA ADOLESCENTES

### 🧭 Introdução

O FriendApp acolhe **adolescentes (14–17 anos)** como parte essencial do ecossistema.

Esta camada garante que eles tenham uma experiência **livre, segura e invisivelmente protegida**, sem se sentirem vigiados ou limitados.

A segurança aqui é **invisível para o usuário, mas técnica e auditável internamente**.

### 🎯 Objetivos da Camada

1. Permitir que adolescentes participem plenamente do app **sem restrição explícita**, mas com filtros invisíveis.
2. Ativar **escudos automáticos** contra perfis abusivos, falsos ou incoerentes.
3. Usar a IA Aurah Kosmos para **adaptar linguagem, feed e conexões** a essa faixa etária.
4. Garantir **proteção contra riscos de abuso, assédio ou manipulação emocional**.

### 🔄 Fluxo Operacional

```
[Cadastro com idade < 18]
↓
[Tela acolhedora: mensagem DUC]
↓
[Verificação DUC via API externa]
↓
- Se verificado → proteção completa ativada
- Se não verificado → proteção invisível reduzida
↓
[Ativação de escudos juvenis invisíveis]
↓
[Feed, Chat e Conexões filtrados em tempo real]
↓
[Painel interno registra eventos para auditoria]
```

### 💛 Texto de Entrada para Menores

👤 Você tem menos de 18 anos?

💛 No FriendApp, a sua experiência é preciosa.

Ao confirmar sua idade com o **DUC (Documento Único de Cidadania)**,  
nosso sistema irá ativar uma **proteção vibracional inteligente**,  
**impedindo que perfis falsos, abusivos ou com intenções tóxicas cheguem até você.**

🌟 Aqui, você está entrando num lugar feito pra ser leve, real e seguro.

🔒 Essa segurança é invisível. Ninguém verá.

Mas ela estará te protegendo o tempo todo.

## Escudos Ativos para Adolescentes

| Escudo                       | Condição Técnica                       | Efeito Aplicado  |
|------------------------------|--|--|
| <b>Escudo Social Juvenil</b> | Ativado para todos < 18                | Feed sem posts intensos, bloqueio de convites de adultos suspeitos |
| <b>Escudo Anti-Assédio</b>   | Linguagem sexual/agressiva detectada   | Mensagem bloqueada + conexão desfeita invisivelmente               |
| <b>Escudo de Conexão</b>     | Adulto não verificado tenta contato    | Convite nunca chega ao menor                                       |
| <b>Escudo de Grupo</b>       | Menor tenta entrar em grupo de adultos | Entrada bloqueada + sugestão de grupos juvenis compatíveis         |

## Fórmula Simplificada de Risco Juvenil

$\text{score\_risco\_juvenil} = (0.4 \times \text{linguagem\_negativa}) + (0.3 \times \text{interacao\_adultos\_suspeitos}) + (0.3 \times \text{comportamento\_incoerente})$   
 $\text{score\_risco\_juvenil} = (0.4 \times \text{linguagem\_negativa}) + (0.3 \times \text{interacao\_adultos\_suspeitos}) + (0.3 \times \text{comportamento\_incoerente})$

$\text{score\_risco\_juvenil} = (0.4 \times \text{linguagem\_negativa}) + (0.3 \times \text{interacao\_adultos\_suspeitos}) + (0.3 \times \text{comportamento\_incoerente})$

- **linguagem\_negativa** = 0 a 1 (peso do NLP em mensagens recebidas).
- **interacao\_adultos\_suspeitos** = +0.3 se tentativa de conexão detectada.
- **comportamento\_incoerente** = divergência entre idade e padrões de uso.

Se **score\_risco\_juvenil**  $\geq 0.6$  → feed e chat entram em **modo silencioso preventivo**.

## Exemplo de Log JSON

```
{
  "usuario_id": "teen-7712",
  "evento": "escudo_juvenil_ativado",
  "detalhes": {
    "tipo": "anti_assedio",
    "conteudo_bloqueado": "mensagem_inadequada",
    "perfil_origem": "adulto_nao_verificado"
  },
  "acao_tomada": "mensagem_bloqueada + conexao_desfeita",
  "score_risco_juvenil": 0.72,
  "timestamp": "2025-09-03T01:15:42Z"
}
```

## Integrações

- **Aurah Kosmos** → ajusta linguagem de interação com menores (mais empática e leve).
- **Feed Sensorial** → bloqueia conteúdos densos ou inadequados.
- **Chat** → ativa filtros de NLP contra assédio.
- **Mapa de Frequência** → marca nós juvenis para proteção coletiva invisível.
- **Painel Interno de Moderação** → exibe tentativas bloqueadas para auditoria ética.

## Fechamento da Camada

O **Sistema de Proteção Invisível para Adolescentes** garante que jovens usem o FriendApp com **liberdade plena e segurança máxima**.



A IA atua como **escudo invisível e educador vibracional**, prevenindo riscos sem limitar a experiência social, mantendo o equilíbrio entre **liberdade, acolhimento e proteção ética**.

## ◆ CAMADA 19 — LÓGICA TÉCNICA DE VERIFICAÇÃO DE IDADE, DOCUMENTOS E DUC/DCO

### Introdução

A verificação de identidade e idade é um dos pontos mais sensíveis do FriendApp.

Nesta camada, definimos a **lógica técnica** de como funcionam as verificações via **DUC (Documento Único de Cidadania)** e **DCO (Documento de Confirmação Opcional)**, garantindo **confiabilidade, privacidade e escalabilidade**.

### Objetivos

1. Garantir que a idade informada no cadastro seja **real e validada**.
2. Oferecer proteção adicional a adolescentes (<18 anos).
3. Reforçar a confiança em conexões autênticas.
4. Evitar que o FriendApp mantenha dados sensíveis localmente, delegando a verificação a APIs especializadas.

### Fluxo Operacional

```
[Usuário envia documento]
↓
[API Gateway]
↓
[verificacao-service]
↓
[API Externa de Identidade (Unico, IdWall)]
↓
Resposta: {status, idade, confiabilidade}
↓
Atualização no perfil do usuário + escudos ativados
```

### Microserviço de Verificação

#### ◆ verificacao-service

- Recebe upload de documento (imagem ou PDF).
- Chama API externa via HTTPS (TLS 1.3).
- Não armazena documento → apenas metadados.
- Retorna score de confiança + idade detectada.

### APIs Externas Recomendadas

Exemplo de integração com API Unico (mock simplificado):

**POST** `/api/verificacao/documento`

```
{
  "usuario_id": "u2025",
  "documento_base64": "..."
}
```

```
}
```

## Resposta

```
{
  "status": "verificado",
  "idade_detectada": 17,
  "score_confianca": 0.92
}
```



## Fórmulas de Confiança

$score\_verificacao = (0.5 \times face\_match) + (0.3 \times ocr\_precisao) + (0.2 \times idade\_consistencia)$   
 $score\_verificacao = (0.5 \times face\_match) + (0.3 \times ocr\_precisao) + (0.2 \times idade\_consistencia)$

$score\_verificacao = (0.5 \times face\_match) + (0.3 \times ocr\_precisao) + (0.2 \times idade\_consistencia)$

- **face\_match** = similaridade selfie x documento.
- **ocr\_precisao** = qualidade da leitura OCR.
- **idade\_consistencia** = idade declarada x idade detectada.

Se `score_verificacao ≥ 0.8` → documento aprovado.



## Proteção de Dados

- Nenhum documento é armazenado permanentemente.
- Logs registram apenas: `status`, `idade_detectada`, `score`.
- Conformidade com **LGPD e GDPR**.
- Dados criptografados com **AES-256** em repouso e **TLS 1.3** em trânsito.



## Logs Técnicos

```
{
  "usuario_id": "u2025",
  "evento": "verificacao_documento",
  "status": "verificado",
  "idade_detectada": 17,
  "score_confianca": 0.92,
  "escudos_ativados": ["juvenil"],
  "timestamp": "2025-09-03T01:42:17Z"
}
```



## Integrações

- **Segurança Vibracional** → ativa escudo juvenil se idade < 18.
- **Aurah Kosmos** → usa confiabilidade para calibrar feedback.
- **Feed Sensorial** → ajusta exibição de conteúdos com base no status.
- **Painel de Confiança** → mostra ao usuário:

“Sua identidade foi confirmada. Seu campo está autenticado.”

## ✓ Fechamento da Camada

A **verificação via DUC/DCO** garante que a identidade seja validada de forma **segura, ágil e transparente**, sem sobrecarregar o FriendApp com dados sensíveis.

Com a lógica clara de integração às APIs externas, fórmulas de score e logs padronizados, os desenvolvedores podem implementar a camada sem dúvidas e em conformidade com a legislação.

## ◆ CAMADA 20 — GATILHOS, REGRAS E LÓGICAS PÓS-CADASTRO

### 🕒 Introdução

Depois que o usuário finaliza o **cadastro e onboarding**, o FriendApp precisa ativar um conjunto de **gatilhos automáticos** que configuram sua jornada inicial.

Esses gatilhos são responsáveis por **sincronizar o perfil com o ecossistema**, aplicar as primeiras proteções, liberar recursos e registrar logs para auditoria.

### 🎯 Objetivos da Camada

1. Garantir que **todos os módulos** recebam os dados iniciais do perfil.
2. Ativar **regras de segurança** específicas conforme score de risco.
3. Automatizar **fluxos pós-cadastro** sem intervenção manual.
4. Registrar logs consistentes para **auditoria e observabilidade**.

### 🔄 Fluxo Operacional

[Cadastro Concluído]  
↓  
[Gatilhos Automáticos Pós-Cadastro]  
↓  
1. Aurah Kosmos ativa leitura contínua  
2. Mapa de Frequência é atualizado no grafo  
3. Feed Sensorial é configurado com filtros  
4. Escudos vibracionais são calibrados  
5. Painel do Usuário exibe status inicial

### 🧩 Gatilhos Ativos

| Gatilho         | Condição Técnica               | Ação Executada   |
|-----------------|--------------------------------|--|
| gatilho_aurah   | Cadastro finalizado            | Inicia leitura contínua e assinatura vibracional         |
| gatilho_feed    | Mapa de Frequência disponível  | Gera feed inicial filtrado por compatibilidade           |
| gatilho_escudos | Score de risco calculado       | Ativa blindagem leve, moderada ou grave                  |
| gatilho_mapa    | Assinatura vibracional gerada  | Posiciona usuário no grafo Neo4j                         |
| gatilho_painel  | Painel de Confiança habilitado | Mostra insights: nível de confiabilidade, escudos ativos |

### 🎲 Regras Técnicas

- **Regra 1 — Score < 0.4** → libera feed completo + conexões expansivas.
- **Regra 2 — 0.4 ≤ Score < 0.75** → feed moderado, conexões limitadas.

- **Regra 3 — Score  $\geq 0.75$**  → feed silencioso, chat restrito, escudo grave ativo.
- **Regra 4 — Verificação DUC aprovada** → reduz risco em -0.3 e libera recursos premium (ex.: Bora, Eventos).

## Logs Técnicos

```
{
  "usuario_id": "u3930",
  "evento": "gatilhos_pos_cadastro",
  "score_risco": 0.52,
  "gatilhos_ativados": [
    "gatilho_aurah",
    "gatilho_feed",
    "gatilho_escudos",
    "gatilho_mapa",
    "gatilho_painel"
  ],
  "acoes": {
    "feed": "moderado",
    "escudo": "moderado",
    "mapa": "cluster_sul_vibracional",
    "painel": "Confiabilidade Média"
  },
  "timestamp": "2025-09-03T02:10:05Z"
}
```

## Integrações

- **Aurah Kosmos** → inicia leitura contínua.
- **Feed Sensorial** → ajusta curadoria com base no score.
- **Mapa Global (Neo4j)** → posiciona usuário em cluster vibracional.
- **Painel de Confiança** → mostra indicadores de segurança e confiabilidade.

## Fechamento da Camada

Os **Gatilhos Pós-Cadastro** são responsáveis por consolidar o perfil vibracional no ecossistema.

Com regras claras, logs padronizados e integração em tempo real, garantem que cada usuário inicie sua jornada no FriendApp com **proteção, transparência e coerência técnica**.

# CAMADA 21 — BANCO DE DADOS UNIFICADO DO SISTEMA DE CADASTRO E PERFIL VIBRACIONAL

## Introdução

O Banco de Dados Unificado é a **coluna vertebral do FriendApp**.

Ele armazena de forma integrada dados de:

- **Cadastro Consciente**
- **Perfil Vibracional**
- **Assinatura Energética**
- **Escudos de Segurança Ativos**

- **Posição no Grafo Global**

A arquitetura é **híbrida** para balancear consistência, performance e escalabilidade.

## **Objetivos da Camada**

1. Centralizar todos os dados de **identidade e vibração** do usuário.
2. Garantir **rápido acesso** a estados em tempo real (feed, escudos, triggers).
3. Permitir **consulta relacional, não relacional e em grafo**.
4. Manter **conformidade com LGPD/GDPR**.

## **Arquitetura do Banco de Dados**

| Banco             | Tipo       | Função   |
|-------------------|------------|--|
| <b>PostgreSQL</b> | Relacional | Cadastro, verificações DUC/DCO, dados estáticos do perfil.           |
| <b>Firestore</b>  | NoSQL      | Estados vibracionais dinâmicos, escudos ativos, trilhas de IA.       |
| <b>Neo4j</b>      | Grafo      | Conexões autênticas, clusters vibracionais, posicionamento coletivo. |

## **Estruturas Técnicas**

### PostgreSQL — Tabela **usuarios\_cadastro**

```
CREATE TABLE usuarios_cadastro (  
  id_usuario UUID PRIMARY KEY,  
  nome TEXT,  
  email TEXT,  
  telefone TEXT,  
  data_nascimento DATE,  
  documento_tipo VARCHAR(10),  
  documento_status VARCHAR(15),  
  score_verificacao FLOAT,  
  data_cadastro TIMESTAMP DEFAULT NOW()  
);
```

### Firestore — Documento **usuarios\_estado\_vibracional**

```
{  
  "usuario_id": "u2025",  
  "assinatura_vibracional": "ae1f28df94b2...",  
  "frequencia_inicial": 7.1,  
  "perfil_vibracional": "Visionário",  
  "grau_coerencia": 0.82,  
  "trilha_aurah": ["expansao_social", "expressao_criativa"],  
  "escudos_ativos": ["explorador"],  
  "status_feed": "moderado",  
  "ultima_atualizacao": "2025-09-03T02:18:42Z"  
}
```

### Neo4j — Nó e Arestas

```
CREATE (:Usuario {  
  id: "u2025",
```

```
    frequencia: 7.1,  
    perfil: "Visionário",  
    coerencia: 0.82  
  })
```

```
MATCH (u:Usuario {id:"u2025"}), (c:Cluster {nome:"sul_vibracional"})  
CREATE (u)-[:PERTENCE]->(c);
```

## Índices e Otimizações

- Índice principal: `id_usuario` (UUID).
- Índices secundários:
  - `score_verificacao` (consultas rápidas de confiabilidade).
  - `perfil_vibracional` (agrupamento de usuários por arquétipo).
  - `frequencia_inicial` (clusterização de proximidade).

## Exemplo de Log de Sincronização

```
{  
  "usuario_id": "u2025",  
  "evento": "sincronizacao_banco",  
  "postgresql": "ok",  
  "firestore": "ok",  
  "neo4j": "ok",  
  "tempo_execucao_ms": 318,  
  "timestamp": "2025-09-03T02:22:55Z"  
}
```

## Integrações

- **Aurah Kosmos** → lê e atualiza frequência + assinatura.
- **Segurança Vibracional** → ativa escudos e grava logs no Firestore.
- **Feed Sensorial** → consulta perfis compatíveis no Neo4j.
- **Painel de Confiança** → exibe dados anonimizados ao usuário.

## Fechamento da Camada

O **Banco Unificado** garante consistência entre **identidade, frequência e conexões**, sustentando toda a lógica vibracional do FriendApp.

Ele é projetado para ser **rápido, seguro e expansível**, permitindo evolução contínua do ecossistema.

# CAMADA 23 — CAMADA DE SEGURANÇA VIBRACIONAL PÓS-CADASTRO (FIREWALLS INTELIGENTES)

## Introdução

Após o cadastro e a ativação inicial de escudos, o usuário entra em operação contínua dentro do FriendApp.

Aqui, a segurança precisa ser mantida por meio de **firewalls vibracionais inteligentes**, que monitoram interações e aplicam bloqueios dinâmicos contra comportamentos nocivos, incoerentes ou fraudulentos.

Os firewalls são **adaptativos, invisíveis e em camadas**, garantindo que a experiência seja livre, mas sempre protegida.

## Objetivos da Camada

1. Atuar como **segunda linha de defesa**, após a blindagem inicial.
2. Monitorar continuamente o uso do feed, chat e conexões.
3. Proteger menores e adultos de abusos, fraudes e incoerências.
4. Registrar todas as intervenções em logs auditáveis.

## Fluxo Operacional

[Usuário em operação contínua]



[Firewalls Inteligentes Ativos]



1. Analisam mensagens, posts e conexões
2. Identificam padrões suspeitos ou incoerentes
3. Aplicam bloqueios invisíveis ou delays
4. Atualizam escudos ativos no Firestore
5. Geram logs técnicos e insights no Painel

## Tipos de Firewalls Vibracionais

| Firewall                        | Condição Técnica   | Ação Executada   |
|---------------------------------|--|--|
| <b>Firewall de Faixa Etária</b> | Usuário <18 + tentativa de conexão com adulto não verificado | Conexão bloqueada invisivelmente                                 |
| <b>Firewall de Linguagem</b>    | NLP detecta abuso, assédio ou ódio                           | Mensagem bloqueada + usuário de origem recebe atraso de resposta |
| <b>Firewall de Flood</b>        | >5 mensagens/min em chat ou feed                             | Delay automático + feed silencioso por 10min                     |
| <b>Firewall de Fraude</b>       | Múltiplos logins suspeitos/VPNs diferentes em 1h             | Conta entra em modo "observação" (restrições temporárias)        |
| <b>Firewall de Dissonância</b>  | Linguagem agressiva em perfil com intenção declarada leve    | Ajusta feed para reduzir conteúdos + alerta para IA Aurah Kosmos |

## Fórmulas de Ativação

Exemplo: **Firewall de Flood**

$$\text{flood\_score} = \frac{\text{mensagens}}{\text{tempo\_min}}$$

$$\text{flood\_score} = \text{tempo\_min} \times \text{mensagens}$$

Se  $\text{flood\_score} > 5$ , firewall é acionado e mensagens novas recebem atraso de 30 segundos.

Exemplo: **Firewall de Linguagem**

$$\text{linguagem\_risco} = \frac{\text{tokens\_negativos}}{\text{tokens\_totais}}$$

$$\text{linguagem\_risco} = \frac{\text{tokens\_negativos}}{\text{tokens\_totais}}$$

Se  $\text{linguagem\_risco} \geq 0.25$  → mensagem é bloqueada.

## Exemplo de Log JSON

```
{
  "usuario_id": "u5511",
  "evento": "firewall_ativado",
  "tipo": "linguagem_abusiva",
  "detalhes": {
    "conteudo": "Você não presta",
    "score_linguagem": 0.32
  },
  "acao_tomada": "mensagem_bloqueada",
  "escudos_ativos": ["linguagem"],
  "timestamp": "2025-09-03T03:02:41Z"
}
```

## Integrações

- **Aurah Kosmos** → reforça escudos e sugere jornada de evolução.
- **Chat Vibracional** → aplica delays ou bloqueios.
- **Feed Sensorial** → filtra posts incoerentes em tempo real.
- **Painel de Confiança** → mostra alertas amigáveis, ex.:  
| "Bloqueamos 1 mensagem incoerente para proteger sua experiência."
- **Observabilidade (Grafana/Prometheus)** → dashboards internos para auditar acionamentos de firewalls.

## Fechamento da Camada

Os **Firewalls Vibracionais Inteligentes** garantem que o FriendApp continue seguro após o cadastro, monitorando linguagem, comportamento e coerência em tempo real.

Eles são **adaptativos, invisíveis e transparentes**, protegendo o usuário enquanto registram logs técnicos que permitem auditoria e evolução constante do sistema.

# CAMADA 24 — APIs DO SISTEMA DE CADASTRO E SINCRONIZAÇÃO COM MICROSERVIÇOS

## Introdução

As **APIs do Sistema de Cadastro** são o ponto central de comunicação entre frontend, backend e microserviços do FriendApp.

Elas permitem que:

- O cadastro consciente funcione sem atritos.
- O perfil vibracional seja processado em tempo real.
- Os dados sejam sincronizados com segurança entre **PostgreSQL, Firestore e Neo4j**.
- Outros módulos (feed, chat, mapa, eventos) recebam os dados já tratados.

## Objetivos da Camada

1. Documentar os **endpoints principais** e seus payloads.
2. Definir **contratos de API claros** para os devs.
3. Garantir **segurança de dados** via autenticação e criptografia.



4. Permitir **integração escalável** com microserviços e APIs externas (verificação de identidade).

## Fluxo Operacional via APIs

```
[Frontend Sensorial]
  ↓ (requisições HTTPS)
[API Gateway - Istio]
  ↓
[Microserviços de Cadastro e Segurança]
  ↓
[Bancos de Dados Híbridos + Aurah Kosmos]
```

## Endpoints Principais

### ◆ Início de Cadastro

**POST** `/api/cadastro/iniciar`

- Cria sessão de cadastro e ID temporário.

Request:

```
{
  "email": "usuario@email.com",
  "nome": "Thayssa",
  "data_nascimento": "1997-07-11"
}
```

Response:

```
{
  "id_temporario": "UUID",
  "status": "iniciado"
}
```

### ◆ Verificação de Documento (DUC/DCO)

**POST** `/api/cadastro/verificar-doc`

- Envia documento para validação em API externa.

Request:

```
{
  "usuario_id": "u2025",
  "documento_base64": "..."
}
```

Response:

```
{
  "status": "verificado",
  "idade_detectada": 17,
  "score_confianca": 0.92
}
```

```
}
```

### ◆ Geração de Perfil Vibracional

**POST** `/api/frequencia/gerar-perfil`

- Processa dados do teste energético e gera mapa inicial.

Request:

```
{
  "usuario_id": "u2025",
  "respostas": { "pergunta_1": "A", "pergunta_2": "B" }
}
```

Response:

```
{
  "frequencia": 7.2,
  "perfil": "Explorador",
  "grau_coerencia": 0.81
}
```

### ◆ Consulta de Status

**GET** `/api/cadastro/status/:id`

- Retorna status atual do fluxo.

Response:

```
{
  "etapa": "teste_energetico",
  "perfil_gerado": false
}
```

## 🧩 APIs Internas Entre Microserviços

| Origem                            | Destino                                    | Evento Sincronizado               |
|-----------------------------------|--|-----------------------------------|
| <code>autenticacao-service</code> | <code>perfil-e-frequencia-service</code>   | Criação de nova conta             |
| <code>verificacao-service</code>  | <code>seguranca-vibracional-service</code> | Resultado da análise de documento |
| <code>aurah-kosmos-core</code>    | <code>feed-sensorial-service</code>        | Assinatura vibracional gerada     |
| <code>perfil-e-frequencia</code>  | <code>conexoes-reais-service</code>        | Ativação de conexões iniciais     |

## 🔒 Segurança das APIs

- **Autenticação:** JWT com validade curta.
- **Criptografia:** TLS 1.3 em trânsito, AES-256 em logs.
- **Rate Limiting:** limite de 100 requests/min por IP.
- **Respostas Padronizadas:** todos os erros seguem formato:

```
{
  "erro": true,
```

```
"codigo": "INVALID_DOC",
"mensagem": "Documento ilegível, tente novamente"
}
```

## Exemplo de Log de Sincronização

```
{
  "usuario_id": "u2025",
  "evento": "api_call",
  "endpoint": "/api/frequencia/gerar-perfil",
  "status": "200",
  "tempo_resposta_ms": 147,
  "payload_size": "2.1kb",
  "timestamp": "2025-09-03T03:15:47Z"
}
```

## Fechamento da Camada

As **APIs do Sistema de Cadastro e Sincronização** são o **núcleo de integração técnica** do FriendApp.

Com contratos bem definidos, segurança robusta e logs estruturados, elas permitem que todo o ecossistema seja **consistente, confiável e escalável**, desde a entrada do usuário até sua plena participação no campo coletivo.

# CAMADA 25 — GATILHOS, EVENTOS E LOGS TÉCNICOS DO SISTEMA DE CADASTRO

## Introdução

Toda ação de segurança vibracional precisa ser **registrada e observável**.

Essa camada define os **gatilhos automáticos** que acontecem durante o cadastro, os **eventos** que são disparados para outros módulos, e o **formato dos logs técnicos** que garantem rastreabilidade, auditoria e evolução contínua do sistema.

## Objetivos da Camada

1. Criar um **pipeline de observabilidade** para cada etapa do cadastro.
2. Garantir que cada ação tomada pela IA Aurah Kosmos seja **auditável**.
3. Fornecer dados para **Painel de Confiança Vibracional** e **dashboards internos**.
4. Manter **consistência entre microserviços** via logs padronizados.

## Gatilhos Automáticos no Cadastro

| Gatilho Técnico                 | Condição de Ativação            | Evento Gerado   |
|---------------------------------|---------------------------------|---|
| <code>gatilho_inicio</code>     | Usuário inicia cadastro         | Cria sessão temporária + log <code>cadastro_iniciado</code> |
| <code>gatilho_doc</code>        | Documento enviado               | Chamada à API externa + log <code>verificacao_doc</code>    |
| <code>gatilho_teste</code>      | Teste de personalidade iniciado | Log <code>teste_iniciado</code> com ID da sessão            |
| <code>gatilho_resultado</code>  | Teste finalizado                | Gera mapa de frequência + log <code>resultado_teste</code>  |
| <code>gatilho_assinatura</code> | Assinatura vibracional gerada   | Log <code>assinatura_gerada</code> + sincronização grafo    |
| <code>gatilho_entrada</code>    | Usuário entra no ecossistema    | Log <code>entrada_vibracional</code> + feed inicial         |

## Eventos Técnicos

Os eventos são enviados pelo **EventBus (Kafka)** e consumidos por microserviços:

| Evento              | Payload Principal                           | Destino Técnico               |
|---------------------|---|-------------------------------|
| novo_usuario_criado | {id_usuario, idade, email}                  | perfil-e-frequencia-service   |
| verificacao_sucesso | {id_usuario, status, score}                 | seguranca-vibracional-service |
| perfil_gerado       | {id_usuario, freq, perfil, coerencia}       | feed-sensorial-service        |
| assinatura_criada   | {id_usuario, hash, cluster}                 | conexoes-reais-service        |
| entrada_confirmada  | {id_usuario, nivel_entrada, escudos_ativos} | painel-vibracional-service    |

## Estrutura dos Logs Técnicos

Todos os logs seguem padrão **JSON**, com campos obrigatórios:

```
{
  "timestamp": "2025-09-03T03:25:17Z",
  "usuario_id": "u8821",
  "evento": "assinatura_gerada",
  "status": "concluido",
  "payload": {
    "frequencia": 71,
    "perfil": "Visionário",
    "grau_coerencia": 0.85,
    "assinatura": "ae1f28df94b2..."
  },
  "origem": "aurah-kosmos-core"
}
```

## Segurança dos Logs

- **Criptografia AES-256** em repouso.
- **Retenção de 12 meses** para logs de auditoria.
- Logs de menores de idade → **anonimizados** (sem nome/email).
- Acesso apenas via **painel interno autenticado (OAuth2 + MFA)**.

## Integrações dos Logs

- **Aurah Kosmos** → retroalimenta IA com feedback real.
- **Painel de Confiança** → traduz logs em insights amigáveis.
- **Grafana/Prometheus** → dashboards internos para devs e moderadores.
- **Suporte** → acesso a logs filtrados para atender apelações de usuários.

## Fechamento da Camada

Os **Gatilhos, Eventos e Logs Técnicos** do cadastro garantem que cada decisão tomada pelo sistema seja:

- **Imediata (gatilhos)**
- **Compartilhada entre módulos (eventos)**
- **Auditável e transparente (logs)**

Essa arquitetura fecha o ciclo de segurança, tornando o FriendApp confiável tanto para usuários quanto para desenvolvedores e investidores.

# CAMADA FINAL — INTEGRAÇÕES E DEPENDÊNCIAS CÍCLICAS DO ECOSSISTEMA

## Introdução

Nenhum módulo do FriendApp funciona de forma isolada.

O **Sistema de Segurança Vibracional** depende e, ao mesmo tempo, alimenta outros sistemas.

Essa camada final documenta todas as **integrações críticas** e as **dependências cíclicas**, para que os devs compreendam claramente:

- De **onde vêm os dados**.
- **Para onde vão os resultados**.
- Quais **gatilhos cruzados** são ativados em tempo real.

## Objetivos da Camada

1. Mapear todas as integrações diretas e indiretas.
2. Explicar o ciclo de dados entre segurança, feed, mapa, conexões e IA.
3. Garantir rastreabilidade total → evitando dúvidas do tipo “*esse sistema depende de quê?*”.
4. Criar base para escalabilidade futura (RA, gamificação, marketplace interno).

## Integrações de Entrada (Sistemas que Alimentam a Segurança)

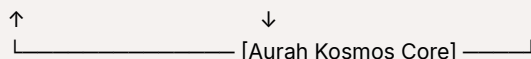
| Sistema Origem             | Tipo de Integração | Dados Fornecidos                                  |
|----------------------------|--------------------|---|
| <b>Cadastro Consciente</b> | API REST           | Idade, documentos (via DUC/DCO), intenção inicial |
| <b>Teste Energético</b>    | API Interna        | Frequência inicial, coerência, arquétipo          |
| <b>Perguntas Rápidas</b>   | Firestore          | Intenção de uso, estado emocional inicial         |
| <b>Aurah Kosmos Core</b>   | EventBus (Kafka)   | Interpretação vibracional, trilhas iniciais       |
| <b>Observabilidade</b>     | Prometheus         | Métricas de performance do fluxo                  |

## Integrações de Saída (Sistemas que Recebem Dados da Segurança)

| Sistema Destino                     | Tipo de Integração    | Ações Disparadas  |
|-------------------------------------|-----------------------|---|
| <b>Feed Sensorial</b>               | API REST              | Filtros de posts, modo silencioso, mensagens educativas     |
| <b>Chat Vibracional</b>             | API Interna           | Delay em mensagens, bloqueio de linguagem incoerente        |
| <b>Mapa Global (Neo4j)</b>          | Neo4j Driver          | Criação de nós e arestas vibracionais                       |
| <b>Conexões Autênticas</b>          | API Interna           | Aprovação ou bloqueio invisível de convites incoerentes     |
| <b>Painel de Confiança</b>          | WebSocket + Firestore | Insights transparentes para o usuário                       |
| <b>Gamificação (futuro)</b>         | API Interna           | Redução/aumento de pontos conforme comportamento energético |
| <b>Realidade Aumentada (futuro)</b> | EventBus              | Dados vibracionais aplicados em camadas RA                  |

## Ciclo de Dependência Cíclica

[Cadastro Consciente] → [Segurança Vibracional] → [Feed/Chat/Mapa/Conexões]



- **Aurah Kosmos** está no centro: valida e retroalimenta todos os módulos.
- **Segurança Vibracional** age como filtro e moderador.
- **Feed e Chat** devolvem dados de comportamento → reprocessados em novos scores.

## Exemplo de Log Integrado

```
{
  "usuario_id": "u5511",
  "evento": "ciclo_integracao",
  "entrada": {
    "cadastro": {"idade": 17, "duc_status": "verificado"},
    "teste": {"freq": 6.8, "perfil": "Empata"},
    "aurah": {"trilha": "cura_emocional"}
  },
  "saida": {
    "feed": "modo_silencioso",
    "chat": "delay_ativado",
    "mapa": "cluster_norte",
    "painel": "Confiabilidade Média"
  },
  "timestamp": "2025-09-03T03:44:28Z"
}
```

## Fechamento da Camada

A **Camada Final** consolida a visão sistêmica: cada módulo depende e retroalimenta outro.

O Sistema de Segurança Vibracional **não é apenas guardião, mas orquestrador**, garantindo que todo o FriendApp opere como um ecossistema coerente, seguro e evolutivo.