

# Documento de Arquitetura de Sistema: LifeSync

## 1. Visão Geral do Produto

O LifeSync é uma plataforma B2B2C focada em bem-estar corporativo, que utiliza gamificação e micro-hábitos para reduzir o burnout e aumentar o engajamento dos colaboradores.

- **Diferencial:** Dashboard de sentimentos em tempo real para o RH (ROI mensurável).

## 2. Funcionalidades Principais

### Para o Colaborador (Mobile App)

- **Autenticação:** Login via SSO corporativo ou Email/Senha.
- **Check-in Emocional:** Registro diário de humor (Mood Tracker) em 3 cliques.
- **Desafios (Micro-hábitos):** Visualização e "check" em tarefas de bem-estar (ex: "Beber 2L de água", "Pausa de 5min").
- **Gamificação:** Sistema de pontuação (XP), níveis e conquistas (Badges).
- **Feed/Ranking:** Visualização do progresso pessoal e ranking anonimizado da equipe.

### Para o RH / Gestor (Web Dashboard)

- **Dashboard Analítico:** Visão agregada do "clima" da empresa (Mapas de calor de estresse/felicidade).
- **Gestão de Usuários:** Cadastro e remoção de colaboradores.
- **Gestão de Campanhas:** Criação de desafios corporativos personalizados.

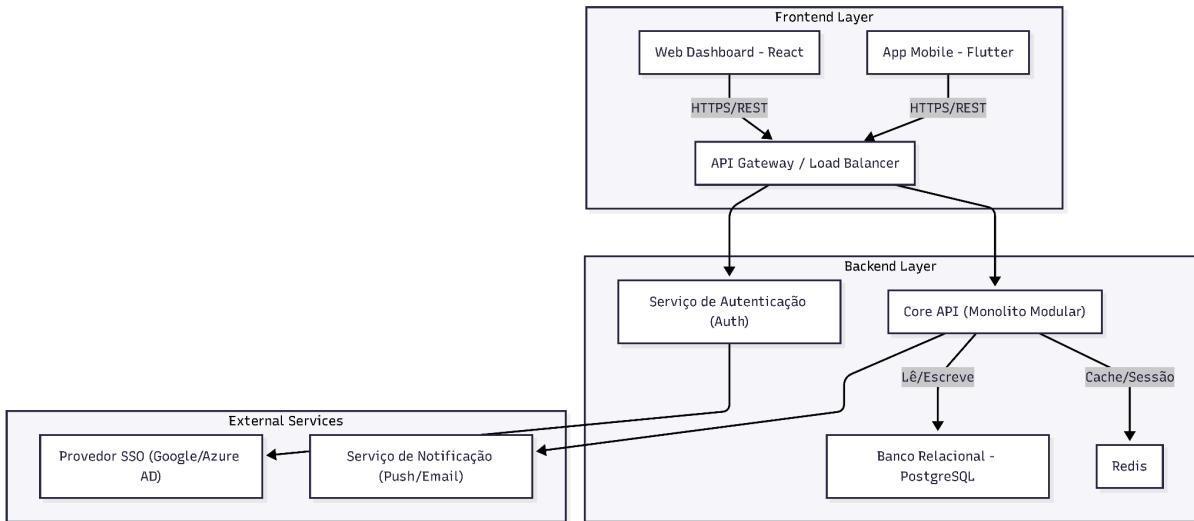
## 3. Tipos de Usuários e Permissões (RBAC)

Perfil	Escopo	Permissões Principais
Colaborador	Mobile	READ / WRITE próprios dados (mood, desafios), READ ranking público.
Gestor RH	Web	READ dados agregados (anônimos) da empresa, WRITE novos desafios/campanhas, MANAGE usuários.
Admin Sistema	Backend	FULL ACCESS configurações globais, MANAGE empresas

		(tenants)
--	--	-----------

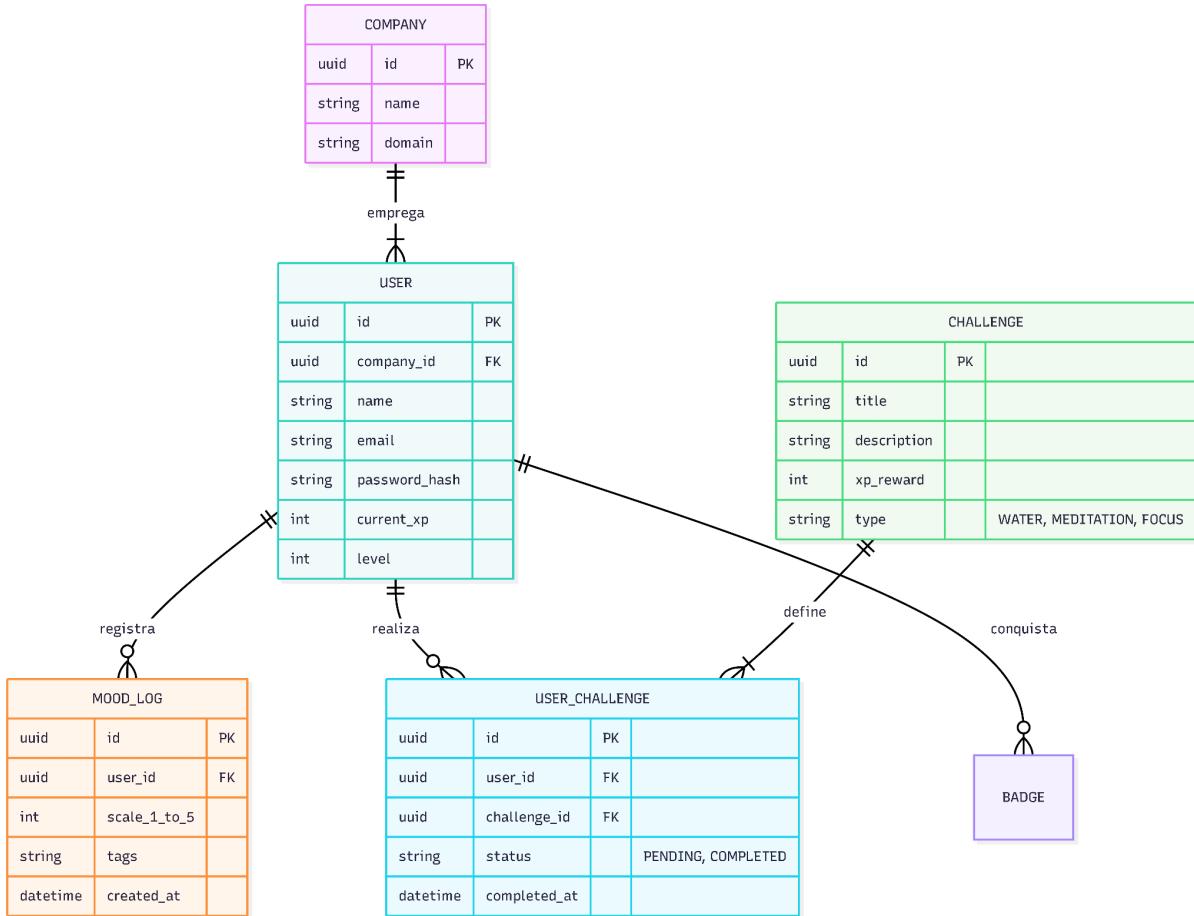
## 4. Diagrama de Arquitetura (High-Level)

A arquitetura segue um padrão de Camadas com API REST, preparada para escalar horizontalmente. O Mobile (Flutter) e a Web (React/Vue) consomem a mesma API.



## 5. Entidades Principais e Relacionamentos

Modelagem focada em rastreabilidade de hábitos e segurança dos dados sensíveis (humor).



## 6. Endpoints da API (Principais Rotas)

### Auth

- POST /auth/login - Autenticação e retorno de JWT.
- POST /auth/register - Registro de usuário (se não for via SSO).

### User Context

- GET /users/me - Retorna perfil, XP e nível do usuário logado.
- GET /users/me/stats - Retorna resumo de dias consecutivos (streak) e badges.

### Gamification & Habits

- GET /challenges - Lista de desafios disponíveis para o dia.
- POST /challenges/{id}/complete - Marca um desafio como feito (Atribui XP).
- POST /mood - Registra o humor do dia (Input: 1-5, Tags).

### RH Analytics (Protegido: Role=HR)

- GET /analytics/mood-overview - Retorna média de humor agrupada por setor/data.
- GET /company/employees - Lista colaboradores ativos.

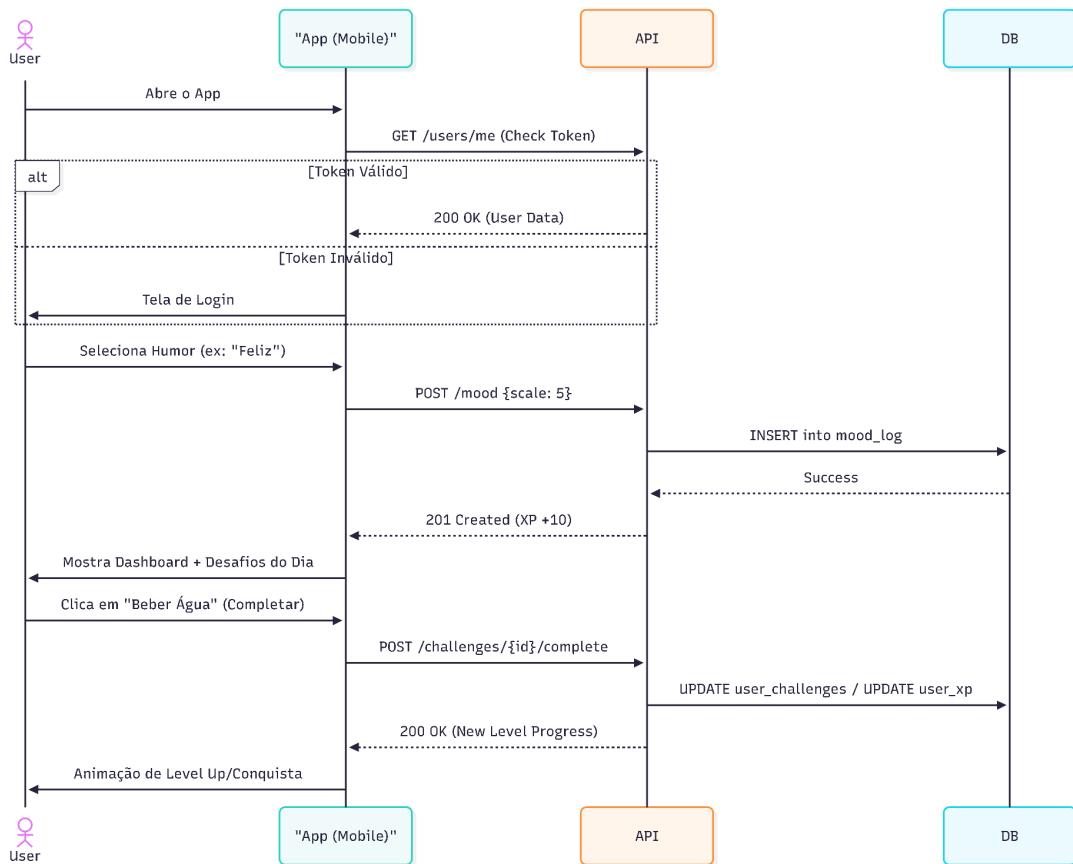
## 7. Tecnologias Sugeridas

Para a implementação em aula, focaremos em uma stack moderna, tipada e com boa documentação:

- Frontend Mobile: Flutter (Dart) - *Seu domínio atual.*
- Frontend Web: React ou Next.js (TypeScript).
- Backend: Node.js com NestJS (TypeScript) ou FastAPI (Python).
- Banco de Dados: PostgreSQL (Robustez relacional) + Prisma ORM (Facilidade de modelagem).
- Infra: Docker (Para subir o banco localmente).

## 8. Fluxo Principal: "Jornada Diária do Usuário"

Este fluxo descreve o "Caminho Feliz" do usuário ao abrir o app pela manhã.



## PROMPTS:

### 1. Prompt de Definição e Arquitetura (Gemini):

"Atue como um Arquiteto de Software Sênior. Preciso desenhar a arquitetura para um sistema de bem-estar corporativo chamado 'LifeSync'. O sistema terá um App Mobile para colaboradores (gamificação, hábitos) e um Dashboard Web para o RH (analytics)."

Liste:

1. Funcionalidades principais para cada perfil.
2. Diagrama de classes/entidades (texto para Mermaid).
3. Stack tecnológica recomendada para um MVP acadêmico.
4. Principais endpoints REST."

### 2. Prompt para Diagramas Visuais (Mermaid):

"Gere um código Mermaid.js representando um diagrama de sequência para o seguinte fluxo: O usuário abre o app, o sistema valida o token, o usuário registra seu humor, o sistema salva no banco e retorna XP. Se o token for inválido, redireciona para login."

### 3. Prompt para Modelagem de Dados:

"Crie um diagrama ER (Entity-Relationship) em formato Mermaid para um sistema de gamificação. Entidades necessárias: Usuário, Empresa, Desafio, Histórico de Desafios e Log de Humor. Inclua chaves primárias e estrangeiras."