

Technical Structure Stellarning

(Frontend, Backend, Blockchain, Infraestrutura e Fluxos)

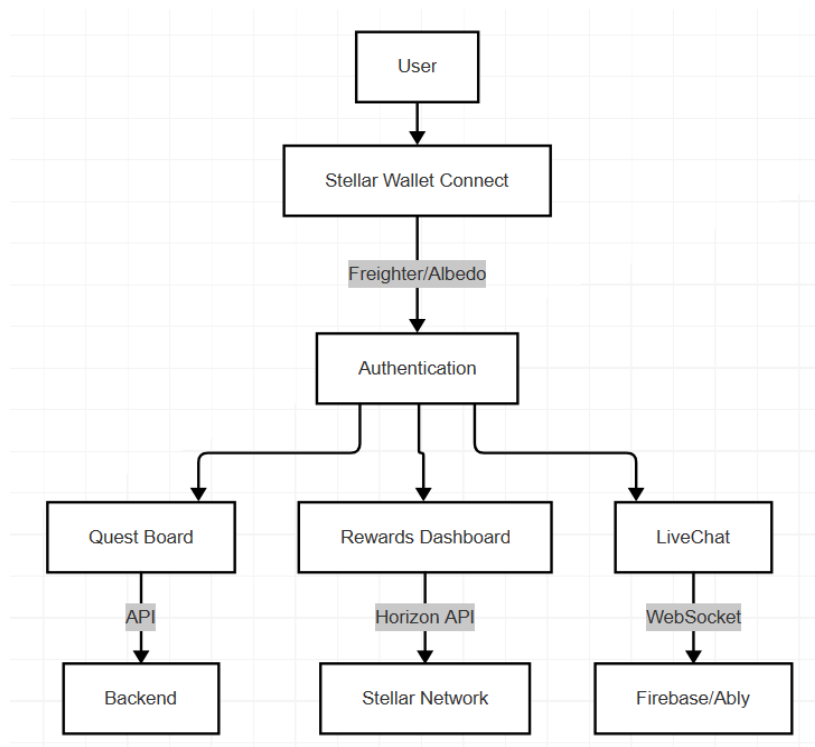
1. Frontend (Next.js + React + Tailwind CSS)

Tecnologias:

- **Framework:** Next.js (SSR/SSG para SEO + rotas API leves).
- **UI:** React + Tailwind CSS (componentes reutilizáveis e design responsivo).
- **Stellar Integration:**
 - @stellar/stellar-sdk: Para interagir com a rede Stellar.
 - @stellar/freighter-api: Login via Freighter Wallet (similar ao MetaMask).
 - albedo-link: Login sem extensão (apenas browser).

Componentes Principais:

- **Stellar Wallet Connect:** Botão de login com Freighter/Albedo.
- **Quest Board:** Lista de missões gamificadas (recompensas em tokens e badges).
- **LiveChat:** Chat global (tempo real).
- **Rewards Dashboard:** Exibe rewards e histórico de transações.



2. Backend (Node.js + Express + Firebase/MongoDB)

Funções Principais:

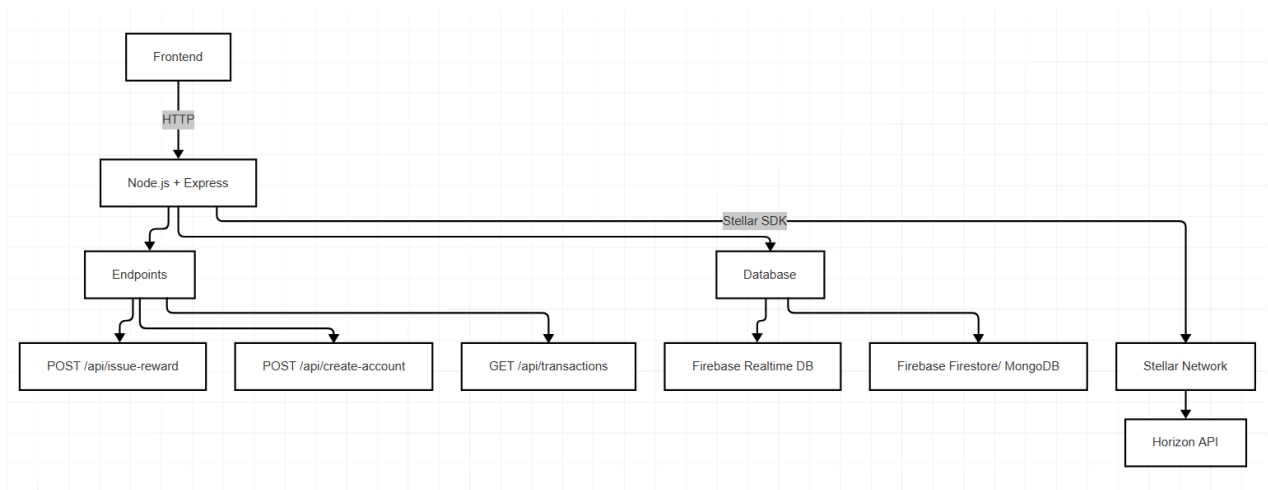
1. **Gerenciar Identidade:**
 - Permitir que novos usuários conectem suas carteiras Stellar à nossa plataforma.
 - Associar chave pública do usuário ao perfil no Firebase/MongoDB.
2. **Emitir Recompensas:**
 - Enviar tokens via Stellar SDK quando o usuário completa uma missão.
3. **Certificação On-Chain:**
 - Registrar conclusão de cursos como transações com memo na Stellar.
4. **Integração com Anchors:**
 - Conectar a anchors como **UltraStellar** para depósitos/saques em stablecoins.

Endpoints (API):

- POST /api/create-account: Criar uma conta Stellar para o usuário.
- POST /api/issue-reward: Envia rewards para uma conta Stellar.
- GET /api/transactions: Retorna histórico de transações do usuário (via Horizon API).

Banco de Dados:

- **Firebase (MVP):**
 - Firestore: Progresso do usuário, missões, inventário.
 - Realtime Database: Chat global e ranking.
- **MongoDB (para escalar na Fase 2):**
 - Armazenar dados complexos (ex: relações entre certificados on-chain e cursos).



3. Blockchain (Rede Stellar)

Configurações:

- **Token:**
 - **Asset Type:** Native Asset (moeda personalizada).
 - **Issuer:** Conta mestra do projeto (chave privada offline).
 - **Supply Control:** Emissão manual (gerenciada pelo backend).
- **Transações:**
 - **Memos:** Usados para vincular ações do app a transações.

Fluxo de Recompensas:

1. Usuário completa uma missão no app.
2. Backend assina uma transação enviando reward para a conta do usuário.
3. Transação inclui um memo descriptografando a ação.
4. Frontend atualiza os rewards do usuário via Horizon API.

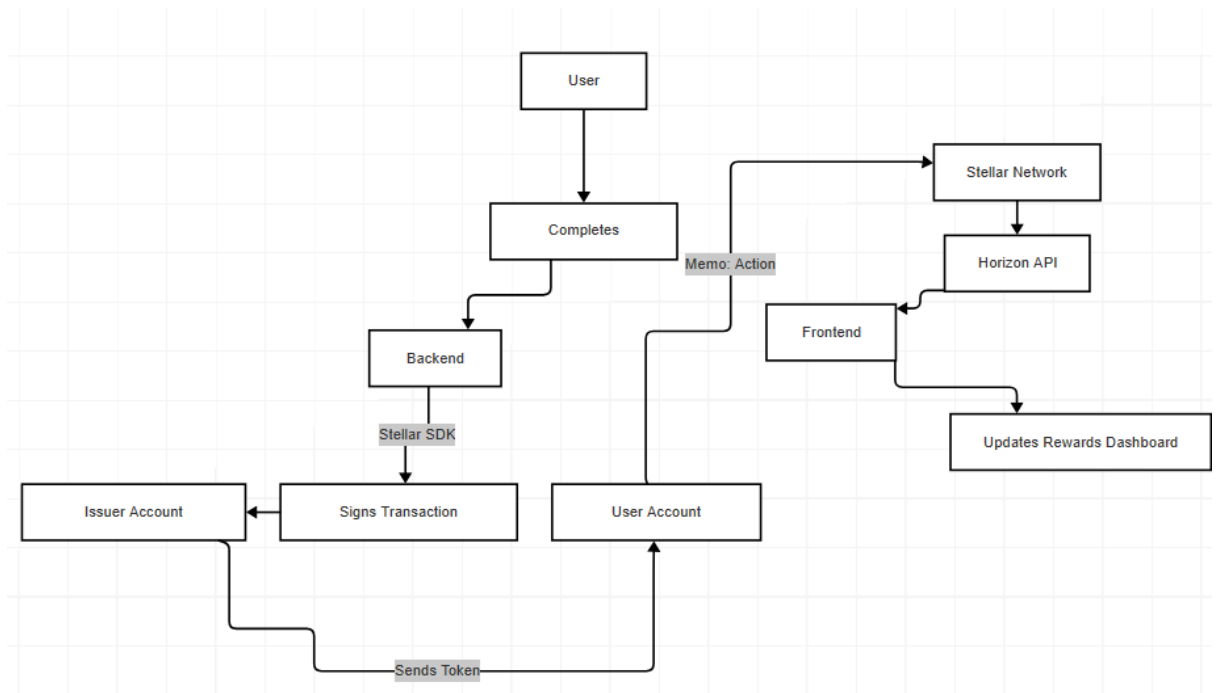
4. Gamificação e Incentivos

Mecânicas:

- **Ranking:** Baseado em tokens acumulados (consulta via Horizon API).
- **Badges NFT-like (certificação):**
 - Representados como transações com memo específico (ex: badge:web3_expert).
 - Exibidos no perfil como "coleccionáveis".
- **Missões (Quests):**
 - Backend valida conclusão e emite recompensas.

Inventário:

- **Itens On-Chain:**
 - Representados por transações com memo (ex: avatar, tokens, certificados).
- **Itens Off-Chain:**
 - Armazenados no Firestore (ex: cores de perfil, acessórios temporários).



5. Infraestrutura e DevOps

Hospedagem:

- **Frontend:** Vercel (deploy automático do Next.js).
- **Backend:** AWS EC2 ou Google Cloud Run (Node.js + Express).
- **Stellar Network:**
 - **Horizon:** Instância pública (<https://horizon.stellar.org>) ou auto-hospedada.

Monitoramento:

- **Sentry:** Para erros no frontend/backend.
- **Stellar Expert:** Explorador de transações em tempo real (ex: <https://stellarexpert.io>).

Segurança:

- **Chaves Privadas:**
 - Issuer do token em cold storage (offline).
 - Multi-signature para contas críticas (ex: 3/5 assinaturas necessárias).
- **Auditoria:**
 - Verificação de transações com ferramentas como **Stellar Laboratory**.

6. Integrações Externas

- **Chat em Tempo Real:** Firebase Realtime Database ou Ably (WebSockets).

