**🧩 Estrutura recomendada para o PRD do seu projeto**

**1. Visão Geral do Produto**

* **Nome do projeto**: Gestão de Visitas AA
* **Objetivo**: Facilitar o agendamento, acompanhamento e finalização de visitas a instituições parceiras, promovendo sobriedade e apoio mútuo.
* **Usuários principais**: Coordenadores de visitas, membros voluntários, administradores do sistema.

**2. Funcionalidades Principais**

| **Módulo** | **Funcionalidade** |
| --- | --- |
| Agenda | Visualizar visitas agendadas por local e data |
| Configurar | Criar nova visita, editar dados, adicionar companheiros |
| Finalizar | Registrar horário real, observações e presença |
| Painel | Estatísticas de visitas, locais e companheiros ativos |

**3. Fluxo de Uso (User Journey)**

* **Configurar visita** → Selecionar local, data, hora, companheiros → Salvar
* **Finalizar visita** → Inserir horário real, observações → Marcar como finalizada
* **Painel** → Visualizar métricas agregadas

**4. Modelagem de Dados (Tabelas Supabase)**

**🗂 Tabela visits**

| **Campo** | **Tipo** | **Descrição** |
| --- | --- | --- |
| id | UUID | Identificador único da visita |
| date | Date | Data agendada |
| time | Time | Hora agendada |
| start\_time | Time | Hora real de início |
| end\_time | Time | Hora real de término |
| notes | Text | Observações |
| isfinalized | Boolean | Status de finalização |
| location\_id | UUID | FK para visit\_locations |

**🗂 Tabela visit\_locations**

| **Campo** | **Tipo** | **Descrição** |
| --- | --- | --- |
| id | UUID | Identificador do local |
| name | Text | Nome do local |
| address | Text | Endereço completo |

**🗂 Tabela companions**

| **Campo** | **Tipo** | **Descrição** |
| --- | --- | --- |
| id | UUID | Identificador do companheiro |
| name | Text | Nome do companheiro |

**🗂 Tabela visit\_companions**

| **Campo** | **Tipo** | **Descrição** |
| --- | --- | --- |
| visit\_id | UUID | FK para visits |
| companion\_id | UUID | FK para companions |

**5. Mapeamento de Código**

* **Hook principal**: useVisits.ts
  + createNewVisit() → Cria visita no Supabase
  + saveVisitChanges() → Atualiza observações e companheiros
  + fetchVisits() → Carrega visitas e normaliza dados
* **Componentes React**:
  + Dashboard.tsx → Exibe visitas agendadas
  + Configurar.tsx → Formulário de criação/edição
  + Finalizar.tsx → Registro de presença e observações

**6. Requisitos Técnicos**

* **Stack**: React + Vite + Supabase
* **Controle de estado**: useState, useEffect
* **Persistência**: Supabase com insert, update, select, delete
* **Autenticação**: (se aplicável) via Supabase Auth ou controle local

**7. Critérios de Aceitação**

* Visita criada deve aparecer na agenda após reload
* Companheiros devem ser persistidos corretamente
* Finalização deve atualizar status e horários reais
* Painel deve refletir métricas atualizadas

**8. Roadmap e Evoluções Futuras**

* 🔒 Autenticação por perfil (admin, voluntário)
* 📱 Versão mobile responsiva
* 📊 Exportação de dados para relatório
* 🔔 Notificações de visita próxima

**🧩 Fases do Ciclo de Vida da Visita**

**🔹 1. Configuração da Visita**

Etapa em que o usuário agenda uma nova visita, seleciona local, data, hora e companheiros.

**Tabelas envolvidas:**

| **Tabela** | **Função** |
| --- | --- |
| visits | Criação do registro principal da visita |
| locations | Fonte de dados para seleção do local (referência lógica via location\_id) |
| companions | Verificação/criação de companheiros |
| visit\_companions | Associação entre visita e companheiros |

**Funções no código:**

* createNewVisit()
* ensureCompanionsExist()

**🔹 2. Finalização da Visita**

Etapa em que o usuário registra os horários reais, observações e confirma os companheiros presentes.

**Tabelas envolvidas:**

| **Tabela** | **Função** |
| --- | --- |
| visits | Atualização dos campos start\_time, end\_time, notes, isfinalized |
| visit\_companions | Recriação da associação com os companheiros reais |
| companions | Verificação/criação de companheiros adicionais |

**Funções no código:**

* saveVisitChanges()
* finalizeVisit()

**🔹 3. Visualização na Agenda**

Exibição das visitas agendadas por local e data, com dados consolidados.

**Tabelas envolvidas:**

| **Tabela** | **Função** |
| --- | --- |
| visits | Fonte principal dos dados |
| locations | Dados do local da visita |
| visit\_companions | Relação entre visita e companheiros |
| companions | Dados dos companheiros |

**Funções no código:**

* fetchVisits()
* normalizeVisit()

**🔹 4. Painel de Métricas**

Exibição de estatísticas como número de visitas, locais diferentes e companheiros ativos.

**Tabelas envolvidas:**

| **Tabela** | **Função** |
| --- | --- |
| visits | Contagem de visitas e locais |
| visit\_companions | Contagem de companheiros únicos |
| companions | Identificação dos companheiros ativos |
| locations | Identificação dos locais distintos |

**Funções no código:**

* Lógica agregada no componente Painel.tsx (ou similar)

**🧠 Visão Geral do Ciclo**

plaintext

[Configurar] → [Finalizar] → [Visualizar na Agenda] → [Painel de Métricas]

↓ ↓ ↓ ↓

visits visits visits visits

locations visit\_companions visit\_companions visit\_companions

companions companions companions companions

visit\_companions