

Documentação do Projeto TMS - MAFFENG

Visão Geral do Projeto

O projeto TMS - MAFFENG consiste em uma solução integrada para gerenciamento de levantamentos técnicos, composta por um aplicativo móvel para técnicos de campo e uma plataforma web para engenheiros administrativos. O objetivo principal é otimizar o processo de coleta de dados em campo e facilitar a visualização, filtragem e gerenciamento dessas informações no escritório.

O aplicativo móvel, desenvolvido com Expo (React Native), permite que os técnicos realizem login seguro, iniciem novos levantamentos associados a um nome, data, gerente responsável e matrícula. Durante o levantamento, os técnicos podem selecionar ambientes específicos (carregados dinamicamente a partir de um arquivo externo, como uma planilha Excel) e registrar informações detalhadas para cada um. Isso inclui a captura de fotos categorizadas em "Vista ampla", "Serviço/itens" e "Detalhes", além da inserção de medidas (altura x largura) ou quantidades para itens pré-definidos (também carregados dinamicamente) e comentários adicionais. Uma funcionalidade crucial é o armazenamento offline, garantindo que os técnicos possam trabalhar mesmo sem conexão à internet, com os dados sendo sincronizados posteriormente quando a conexão for restabelecida. As fotos capturadas são automaticamente enviadas para o Supabase Storage.

A aplicação web, construída com Next.js e estilizada com TailwindCSS (utilizando a paleta de cores corporativa: preto, vermelho, branco e azul piquet), oferece um painel administrativo para os engenheiros. Após o login seguro, os engenheiros têm acesso a uma interface inspirada no Google Drive, com visualização dos levantamentos organizados em uma estrutura de pastas (por exemplo, por técnico, ano, mês). Uma barra lateral permite filtrar os levantamentos por técnico, intervalo de datas e ambiente específico. A visualização detalhada de cada levantamento permite analisar as informações coletadas, incluindo as fotos e medidas. A funcionalidade de exportação de dados em formatos como Excel ou PDF foi planejada, mas sua implementação foi adiada para uma fase futura, conforme solicitado.

Arquitetura da Solução

A arquitetura do TMS - MAFFENG é baseada em um backend centralizado no Supabase, que fornece serviços de banco de dados (PostgreSQL), autenticação (Supabase Auth) e armazenamento de arquivos (Supabase Storage). Essa escolha simplifica a infraestrutura e oferece uma plataforma escalável e integrada.

Backend (Supabase):

- * **Banco de Dados (PostgreSQL):** Armazena todos os dados estruturados, como informações dos usuários (técnicos, engenheiros), detalhes dos levantamentos (nome, data, gerente, matrícula), informações sobre ambientes e itens (potencialmente sincronizados do arquivo externo), e os dados coletados (medidas, quantidades, comentários, referências às fotos).
- * **Autenticação (Supabase Auth):** Gerencia o login e a segurança tanto para o aplicativo móvel quanto para a aplicação web, garantindo que apenas usuários autorizados acessem o sistema.
- * **Armazenamento (Supabase Storage):** Utilizado para armazenar as fotos capturadas pelos técnicos no aplicativo móvel. As referências (URLs) dessas fotos são salvas no banco de dados PostgreSQL, associadas ao respectivo levantamento e ambiente.

Frontend Mobile (Expo / React Native):

- * Interface de usuário nativa para Android e iOS.
- * Componentes React Native para telas de login, início de levantamento, seleção de ambiente, captura de fotos (utilizando `expo-camera` ou `expo-image-picker`), entrada de dados e visualização de informações.
- * Lógica para carregar dados dinâmicos (ambientes, itens) de um arquivo (provavelmente Excel, usando uma biblioteca como `xlsx`).
- * Gerenciamento de estado (possivelmente com Context API, Zustand ou Redux Toolkit) para dados do levantamento e estado da aplicação.
- * Implementação de armazenamento local (por exemplo, usando AsyncStorage ou SQLite via `expo-sqlite`) para funcionalidade offline.
- * Lógica de sincronização para enviar dados locais para o Supabase quando online.
- * Integração com Supabase SDK (JavaScript) para autenticação, interação com banco de dados e upload de arquivos para o Storage.

Frontend Web (Next.js / React / TailwindCSS):

- * Interface de usuário web responsiva.
- * Páginas renderizadas no servidor (SSR) ou estaticamente (SSG) com Next.js para performance e SEO (embora SEO não seja o foco principal para um painel interno).
- * Componentes React para layout, login, painel de controle, barra lateral de filtros e visualização dos levantamentos (incluindo a estrutura de pastas).

- * Estilização com TailwindCSS, utilizando um tema customizado com as cores corporativas.
- * Integração com Supabase SDK (JavaScript) para autenticação e busca de dados do banco de dados para exibição no painel.
- * Gerenciamento de estado para filtros e dados exibidos.

Fluxo de Dados:

1. **Configuração Inicial:** Um arquivo (Excel) contendo a lista de ambientes e itens (com suas propriedades: nome, descrição, se requer medida ou quantidade) é preparado e disponibilizado (seja no próprio app ou em local acessível).
2. **Login (Mobile/Web):** Usuários se autenticam via Supabase Auth.
3. **Criação do Levantamento (Mobile):** Técnico inicia um novo levantamento, inserindo dados iniciais. App carrega listas de ambientes e itens do arquivo.
4. **Coleta de Dados (Mobile):** Técnico seleciona ambiente, tira fotos (upload para Supabase Storage), insere medidas/quantidades/comentários para os itens. Dados são salvos localmente (offline).
5. **Sincronização (Mobile):** Quando online, o app sincroniza os dados locais (detalhes do levantamento, referências das fotos, medidas, etc.) com o banco de dados Supabase.
6. **Visualização (Web):** Engenheiro loga no painel web. A aplicação busca os dados dos levantamentos (já sincronizados) do Supabase.
7. **Filtragem (Web):** Engenheiro utiliza filtros na barra lateral. A aplicação refaz a busca no Supabase com os critérios de filtro.
8. **Navegação (Web):** Engenheiro navega pela estrutura de pastas e visualiza detalhes de um levantamento específico.

Estrutura do Projeto

O projeto está organizado em um monorepositório (embora não configurado formalmente como tal aqui) com duas pastas principais: `mobile` e `web`.

```
/tms-maffeng
|-- /mobile          # Projeto Expo (React Native)
|   |-- /assets      # Fontes, imagens estáticas
|   |-- /data        # Arquivos de dados (ex: planilha de ambientes/itens)
|   |-- /src
|       |-- /components  # Componentes reutilizáveis (botões, inputs, etc.)
|       |-- /navigation  # Configuração da navegação (React Navigation)
|       |-- /screens     # Telas da aplicação (Login, Home, NewSurvey, etc.)
|       |-- /services    # Lógica de integração (Supabase API, Leitura de Arquivo)
|       |-- /store       # Gerenciamento de estado (Context, Zustand, Redux)
|       |-- /utils       # Funções utilitárias
|-- app.json         # Configuração do Expo
|-- package.json     # Dependências e scripts
```

```

| | -- ...          # Outros arquivos de configuração (babel.config.js, etc.)
|
| -- /web           # Projeto Next.js
| | -- /app         # Diretório App Router (Next.js 13+)
| | | -- /dashboard # Rota e página do painel principal
| | | -- /login     # Rota e página de login
| | | -- layout.js   # Layout raiz da aplicação
| | | -- page.js     # Página inicial (pode redirecionar para login/dashboard)
| | -- /components  # Componentes React reutilizáveis (Sidebar, SurveyView,
etc.)
| | -- /lib          # Funções utilitárias, clientes API (Supabase)
| | -- /public       # Arquivos estáticos (imagens, favicons)
| | -- /styles       # Arquivos CSS globais (globals.css)
| | -- next.config.js # Configuração do Next.js
| | -- tailwind.config.js # Configuração do TailwindCSS (com cores customizadas)
| | -- package.json  # Dependências e scripts
| | -- ...           # Outros arquivos de configuração (.eslintrc.json, etc.)
|
|-- README.md        # Este arquivo de documentação
|-- todo.md          # Lista de tarefas (para desenvolvimento interno)

```

Configuração e Instalação

Instruções detalhadas sobre como configurar o ambiente de desenvolvimento para as partes mobile e web serão adicionadas posteriormente. Isso incluirá a instalação de dependências (Node.js, Expo CLI, Yarn/NPM), configuração das variáveis de ambiente para conexão com o Supabase (URL do projeto e Chave Pública Anon) e instruções para rodar cada aplicação localmente.

(Continua...)

Configuração e Instalação

Para configurar o ambiente de desenvolvimento e executar as aplicações localmente, siga os passos abaixo. É necessário ter o Node.js (versão LTS recomendada) e o Yarn ou NPM instalados.

Configuração do Supabase:

1. Crie um projeto no [Supabase](#).
2. No painel do seu projeto Supabase, vá até "Project Settings" > "API". Anote a "Project URL" e a "anon public" key.
3. Configure as tabelas necessárias no banco de dados PostgreSQL. Isso incluirá tabelas para usuários (gerenciada pelo Supabase Auth), levantamentos, ambientes, itens,

detalhes dos itens coletados e referências de fotos.

4. Configure o Supabase Storage para permitir o upload de imagens pelos técnicos.

Configuração do Projeto Mobile (Expo):

1. Navegue até o diretório `mobile` : `cd /home/ubuntu/tms-maffeng/mobile` .

2. Instale as dependências: `yarn install` ou `npm install` .

3. Crie um arquivo de configuração de ambiente (por exemplo, `.env`) na raiz do diretório `mobile` para armazenar as credenciais do Supabase:

```
EXPO_PUBLIC_SUPABASE_URL=SUA_SUPABASE_URL
```

```
EXPO_PUBLIC_SUPABASE_ANON_KEY=SUA_SUPABASE_ANON_KEY
```

4. Certifique-se de que o arquivo de dados (Excel/CSV/TXT) com ambientes e itens esteja na pasta `/mobile/data` e que a lógica em `/mobile/src/services` esteja configurada para lê-lo corretamente.

5. Para executar o aplicativo em um emulador ou dispositivo físico, utilize o Expo CLI: `npx expo start` . Siga as instruções no terminal para abrir o app.

Configuração do Projeto Web (Next.js):

1. Navegue até o diretório `web` : `cd /home/ubuntu/tms-maffeng/web` .

2. Instale as dependências: `yarn install` ou `npm install` .

3. Crie um arquivo `.env.local` na raiz do diretório `web` para as credenciais do Supabase:

```
NEXT_PUBLIC_SUPABASE_URL=SUA_SUPABASE_URL
```

```
NEXT_PUBLIC_SUPABASE_ANON_KEY=SUA_SUPABASE_ANON_KEY
```

4. Execute o servidor de desenvolvimento: `yarn dev` ou `npm run dev` . Acesse a aplicação no navegador através do endereço `http://localhost:3000` (ou a porta indicada no terminal).

Instruções de Uso (Mobile)

O aplicativo móvel é destinado aos técnicos de campo para registrar novos levantamentos.

1. **Login:** Abra o aplicativo e insira suas credenciais (email e senha) gerenciadas pelo Supabase Auth.
2. **Tela Inicial:** Após o login, você verá a tela inicial. Toque em "Iniciar Novo Levantamento".
3. **Novo Levantamento:** Preencha o nome do levantamento, seu nome de gerente e sua matrícula. A data será preenchida automaticamente. Toque em "Selecionar Ambiente".
4. **Seleção de Ambiente:** Uma lista de ambientes disponíveis (carregada do arquivo de dados) será exibida. Toque no ambiente que deseja inspecionar.

5. **Detalhes do Levantamento (Ambiente):** Nesta tela, você registrará os dados para o ambiente selecionado:
 - **Fotos:** Toque nos botões "Tirar Foto - Vista Ampla", "Tirar Foto - Serviço/Itens" ou "Tirar Foto - Detalhes" para abrir a câmera e capturar as imagens correspondentes. As fotos serão enviadas ao Supabase Storage.
 - **Itens e Medidas:** Uma lista de itens pré-definidos para o ambiente será exibida. Para cada item, insira a quantidade ou as medidas (altura x largura), conforme aplicável. Adicione comentários relevantes no campo correspondente.
6. **Concluir Ambiente:** Após registrar todas as informações e fotos para o ambiente atual, toque em "Concluir Ambiente". Você pode ser direcionado de volta à tela de seleção para escolher outro ambiente ou finalizar o levantamento.
7. **Sincronização:** Os dados são salvos localmente. O aplicativo tentará sincronizar os dados com o servidor Supabase automaticamente quando houver conexão com a internet.

Instruções de Uso (Web)

A plataforma web é destinada aos engenheiros administrativos para visualizar e gerenciar os levantamentos.

1. **Login:** Acesse a URL da aplicação web e faça login com suas credenciais.
2. **Painel Principal:** Você será direcionado ao painel principal, que exibe os levantamentos.
3. **Navegação/Visualização:** A visualização principal organiza os levantamentos em uma estrutura de pastas (por exemplo, por técnico/ano/mês). Navegue pelas pastas para encontrar levantamentos específicos. Clique em um levantamento para ver seus detalhes (informações gerais, ambientes inspecionados, fotos, medidas, comentários).
4. **Filtros:** Utilize a barra lateral para filtrar os levantamentos exibidos. Você pode filtrar por nome do técnico, intervalo de datas e ambiente específico.
5. **Exportação (Futuro):** A funcionalidade de exportar dados para Excel ou PDF será implementada em uma fase futura.

Próximos Passos / Considerações Futuras

Este projeto fornece uma base sólida para o sistema TMS - MAFFENG. Algumas áreas para desenvolvimento futuro incluem:

- **Implementação Completa da Autenticação:** Integrar completamente o Supabase Auth nas telas de login mobile e web.

- **Leitura Dinâmica de Arquivos (Mobile):** Implementar a lógica robusta para ler e processar o arquivo de ambientes e itens (Excel, CSV, etc.).
- **Lógica Offline e Sincronização (Mobile):** Desenvolver o mecanismo completo de armazenamento offline e sincronização de dados com tratamento de conflitos, se necessário.
- **Câmera e Armazenamento (Mobile):** Integrar expo-camera ou expo-image-picker e finalizar o fluxo de upload para o Supabase Storage.
- **Busca de Dados e Filtros (Web):** Conectar a interface web ao Supabase para buscar dados reais e implementar a lógica de filtragem e paginação/scroll infinito.
- **Visualização em Pastas/Árvore (Web):** Refinar a visualização dos levantamentos para uma estrutura de pastas/árvore interativa.
- **Funcionalidade de Exportação (Web):** Implementar a exportação de dados para Excel e PDF, conforme solicitado originalmente.
- **Testes:** Adicionar testes unitários, de integração e end-to-end.
- **Deploy:** Configurar o deploy contínuo para o mobile via EAS Build e para a web via Vercel.
- **Melhorias de UI/UX:** Refinar a interface do usuário e a experiência geral com base no feedback dos usuários.