

Documentação de Progresso do Projeto ModularCompany Mobile

Visão Geral do Projeto

O ModularCompany Mobile é um aplicativo desenvolvido com React Native e Expo que consome a API do ModularCompany para oferecer funcionalidades similares à plataforma web, adaptadas para dispositivos móveis. O aplicativo foca inicialmente no módulo de Controle de Horas, permitindo o registro e gerenciamento de horas trabalhadas pelos funcionários.

Estrutura de Usuários

O sistema possui quatro níveis de acesso, cada um com permissões e funcionalidades específicas:

1. **Desenvolvedor:** Responsável por cadastrar empresas e gerenciar acesso e pagamentos
2. **Administrador:** Gerencia a empresa, seleciona módulos, cadastra gerentes e funcionários
3. **Gerente:** Gerencia funcionários e tem acesso a relatórios
4. **Funcionário:** Acessa os serviços específicos para funcionários



Estado Atual do Desenvolvimento

Implementado (☑)




- **Sistema de Autenticação** (☑)
 - Tela de login funcional com validações
 - Armazenamento seguro de token e dados do usuário com AsyncStorage
 - Redirecionamento para dashboard após login bem-sucedido
 - Verificação de autenticação em rotas protegidas
- **Dashboard Inicial** (☑)
 - Exibição de informações básicas do usuário
 - Menu de acesso rápido para principais funcionalidades
 - Funcionalidade de logout
- **Navegação Bottom Tab** (☑)
 - Implementação da barra de navegação inferior
 - Adaptação das abas para cada tipo de usuário
- **Módulo de Controle de Horas** (☑)
 - Listagem de registros de horas com controle de acesso por perfil:
 - Administradores e Desenvolvedores: visualizam todas as horas da empresa
 - Gerentes: visualizam horas dos gerentes e funcionários
 - Funcionários: visualizam apenas suas próprias horas

- Formulário para criação de novos registros com validação
- Filtros por status (pendente, aprovado, rejeitado)
- Barra de pesquisa para filtrar por nome de usuário ou projeto
- Aprovação/Rejeição de registros com feedback visual
- Interface responsiva e adaptada para diferentes perfis de usuário

Em Desenvolvimento ()

- **Telas por Tipo de Usuário** ()
 - Telas específicas para Desenvolvedor
 - Telas específicas para Administrador
 - Telas específicas para Gerente
 - Telas específicas para Funcionário
- **Gerenciamento de Usuários** ()
 - Listagem de usuários
 - Adição de novos usuários
 - Edição de usuários existentes

Pendente ()

- **Perfil de Usuário** ()
 - Visualização de dados pessoais
 - Edição de informações básicas
 - Alteração de senha
- **Relatórios** ()
 - Visualização de relatórios de horas
 - Filtros por período e funcionário
 - Exportação de relatórios
- **Notificações** ()
 - Sistema de notificações dentro do app
 - Notificações push para eventos importantes

Componentes e Telas

Componentes Implementados

1. Login Screen - Tela de login com:

- Campo de email e senha
- Validação de formulário
- Botão de login com indicador de carregamento
- Link para recuperação de senha

2. **Dashboard Screen** - Tela inicial após login com:

- Cabeçalho com informações do usuário
- Cards de acesso rápido
- Botão de logout

3. **Time Entries Screen** - Tela de controle de horas com:

- Lista de registros com indicação visual de status (aprovado, pendente, rejeitado)
- Filtros por status para facilitar visualização
- Barra de pesquisa para localizar registros específicos (visível apenas para admin/manager)
- Formulário para criação de novos registros com validação
- Modal de aprovação/rejeição com campos para justificativa
- Controle de permissões baseado no perfil do usuário

4. **Users Screen** - Tela de gerenciamento de usuários

Melhorias Recentes Implementadas

1. **Controle de Acesso Hierárquico:**

- Implementação de chamadas à API com parâmetros específicos por perfil para buscar registros apropriados
- Administradores e Desenvolvedores visualizam todos os registros da empresa usando `?all=true`
- Gerentes visualizam registros de gerentes e funcionários usando `?managedUsers=true`
- Funcionários visualizam apenas seus próprios registros (comportamento padrão da API)

2. **Interface de Aprovação de Horas:**

- Modal detalhado para visualizar informações do registro
- Interface para aprovar ou rejeitar registros com feedback visual
- Campo obrigatório para informar motivo de rejeição
- Atualização em tempo real da lista após aprovação/rejeição

3. **Pesquisa e Filtragem Avançada:**

- Barra de pesquisa para localizar registros por nome de usuário ou projeto
- Filtros múltiplos combinando status e texto de pesquisa
- Interface adaptativa que mostra ferramentas relevantes apenas para usuários autorizados

4. **Apresentação Melhorada de Usuários:**

- Exibição clara do nome e cargo do usuário para cada registro
- Tags visuais indicando o cargo (Administrador, Gerente, Funcionário)
- Formatação melhorada para facilitar a leitura de múltiplos registros

Próximos Componentes a Implementar

1. **Tela de Relatórios** com:

- Resumo de horas trabalhadas por período
- Gráficos e estatísticas
- Opções de exportação

2. Tela de Perfil Completa com:

- Foto de perfil
- Informações detalhadas
- Formulário para edição de dados

Próximos Passos

1. Finalizar as telas específicas para cada tipo de usuário
2. Implementar a tela de relatórios e análises
3. Desenvolver o perfil de usuário completo
4. Adicionar sistema de notificações

Notas Técnicas

API

- Base URL: <https://modularcompany.vercel.app/api>
- Endpoints principais:
 - Login: [/mobile-auth](#) (POST)
 - Perfil: [/mobile-profile](#) (GET)
 - Registros de Horas: [/mobile-time-entries](#) (GET, POST)
 - Aprovação de Horas: [/mobile-time-entries/:id/approve](#) (PUT)

Dependências Principais

- expo-router - Sistema de navegação
- axios - Requisições HTTP
- @react-native-async-storage/async-storage - Armazenamento local
- react-native-safe-area-context - Gestão de áreas seguras
- @expo/vector-icons - Ícones
- date-fns - Manipulação de datas
- @react-native-community/datetimepicker - Seletor de data e hora

Estrutura de Arquivos

```

app/
├── _layout.tsx           # Layout principal
├── auth/                # Telas de autenticação
│   ├── _layout.tsx
│   └── login.tsx
├── (dashboard)/         # Telas de dashboard
│   ├── _layout.tsx
│   └── index.tsx

```

```
|— (tabs)/                                # Navegação por abas
|   |— _layout.tsx
|   |— index.tsx                          # Home tab
|   |— time-entries.tsx                   # Tab de registros de horas
|   |— users.tsx                         # Tab de gerenciamento de usuários
|   |— companies.tsx                     # Tab de gerenciamento de empresas
|   |— reports.tsx                       # Tab de relatórios
|   |— profile.tsx                       # Tab de perfil
assets/
|— images/                               # Imagens
components/                             # Componentes reutilizáveis
constants/
|— Colors.ts                             # Cores do tema
hooks/                                  # Hooks customizados
docs/
|— API_MOBILE.md                         # Documentação da API Mobile
|— ProjectProgress.md                   # Documentação de progresso do projeto
```