

# APP LG - Clube Gym Pass

## Resumo Completo das Funcionalidades

### Visão Geral do Projeto






- **Nome:** APP LG (Clube Gym Pass)
  - **Tecnologia:** Next.js 14 + React 18 + TypeScript + Tailwind CSS
  - **Backend:** Supabase (Database, Auth, Storage, Edge Functions)
  - **Pagamentos:** MercadoPago
  - **Funcionalidades Especiais:** QR Code scanning, JWT tokens, real-time updates
- 



## 1. Sistema de Autenticação

**Localização:** `/app/(auth)/login/page.tsx`

### Funcionalidades:

-  Login via "Magic Link" (email sem senha)
-  Autenticação através do Supabase Auth
-  Redirecionamento automático para dashboard após login
-  Persistência de sessão
-  Sistema de logout

**Tecnologias:** Supabase Auth, OTP via email

---



## 2. Dashboard do Membro

**Localização:** `/app/dashboard/page.tsx`

**Funcionalidades:**

- ☒ Visualização do plano atual do usuário
- ☒ Status da assinatura (ativo/inativo/pendente)
- ☒ Preços formatados em reais (R\$)
- ☒ Botão para realizar pagamento quando necessário
- ☒ Link para acessar carteirinha digital
- ☒ Feedback visual de status de pagamento
- ☒ Tratamento de erros de pagamento

**Integração:** Supabase (tabelas: subscriptions, plans)

---



### 3. Carteirinha Digital (Wallet)

**Localização:** `/app/wallet/page.tsx`

**Funcionalidades:**

- ☒ Geração de QR Code dinâmico
- ☒ Token JWT renovado automaticamente a cada 55 segundos
- ☒ Exibição visual do QR Code (240px)
- ☒ Preview do token (primeiros 18 caracteres)
- ☒ Auto-refresh para manter token válido

**Tecnologias:** QRCode library, JWT tokens, Canvas API

---



### 4. Scanner de Check-in (Academias)






**Localização:** `/app/academias/scanner/page.tsx`

**Funcionalidades:**

- ☒ Leitura de QR Code via câmera do dispositivo
- ☒ Validação de tokens por unidade específica
- ☒ Sistema de cooldown entre check-ins
- ☒ Controle de limite mensal de visitas
- ☒ Verificação de status da assinatura
- ☒ Feedback visual de sucesso/erro
- ☒ Mensagens personalizadas para diferentes tipos de erro

**Tecnologias:** html5-qrcode library, Camera API






**Estados de Validação:**

-  Check-in bem-sucedido
  -  Cooldown ativo
  -  Limite mensal atingido
  -  Assinatura inativa
  -  Token inválido
- 

## 5. Painel Administrativo

**Localização:** `/app/admin/page.tsx`

**Funcionalidades:**

-  Listagem de filiais (organizações)
-  Gestão de unidades/academias
-  Controle de usuários registrados
-  Status de aprovação de usuários
-  Layout responsivo com cards








**Tabelas Supabase:** orgs, units, profiles

---

## 6. Sistema de Tickets/Suporte

**Localização:** `/app/tickets/page.tsx`

**Funcionalidades:**

-  Abertura de novos tickets de suporte
-  Formulário com assunto e descrição
-  Listagem de tickets existentes
-  Sistema de mensagens por ticket
-  Status de tickets (aberto/fechado)
-  IDs únicos para identificação
-  Histórico de mensagens por ticket

**Tabelas Supabase:** tickets, ticket\_messages

---



## 7. CRM - Gestão de Leads

**Localização:** `/app/crm/leads/page.tsx`

**Funcionalidades:**

- ☒ Formulário de captura de leads
- ☒ Campos: Nome, Email, Telefone, Fonte
- ☒ Tabela de visualização de leads capturados
- ☒ Status de leads (novo, convertido, etc.)
- ☒ Ordenação por data de criação
- ☒ Interface responsiva

**Fonte padrão:** "site"

---



## 8. Sistema de Pagamentos

**Localização:** `/app/api/payments/checkout/route.ts`

**Funcionalidades:**

- ☒ Integração com MercadoPago
- ☒ Geração de links de pagamento
- ☒ Suporte a sandbox e produção
- ☒ Redirecionamento automático
- ☒ Tratamento de erros de pagamento
- ☒ Webhooks para confirmação (configurável)

**Variáveis de ambiente:** MP\_ACCESS\_TOKEN, MP\_WEBHOOK\_URL

---



## 9. APIs e Backend

**Supabase Integration:**

- ☒ Configuração de cliente e servidor
- ☒ Tratamento de ambientes sem configuração
- ☒ Fallback graceful quando Supabase não configurado

-  Persistência de sessão

## Estrutura de Tabelas (Schema):








- `profiles` - Perfis de usuários
- `subscriptions` - Assinaturas ativas
- `plans` - Planos disponíveis
- `orgs` - Filiais/Organizações
- `units` - Unidades/Academias
- `tickets` - Tickets de suporte
- `ticket_messages` - Mensagens dos tickets
- `leads` - Leads capturados pelo CRM

## APIs Específicas:

- `/api/payments/checkout` - Checkout do MercadoPago
  - `/api/wallet/issue` - Emissão de tokens da carteirinha
  - `/api/checkins/validate` - Validação de check-ins
  - `/api/webhooks/payment` - Webhooks de pagamento
- 

## 10. Interface e UX

### Características:

-  Design responsivo com Tailwind CSS
-  Layout limpo e moderno
-  Navegação intuitiva no header
-  Cards para organização de conteúdo
-  Badges para status visuais
-  Botões com estados de loading
-  Feedback visual para ações do usuário

### Componentes Principais:

- Header com navegação
  - Cards informativos
  - Formulários responsivos
  - Tabelas organizadas
  - Badges de status
- 



## 11. Segurança e Tokens

### JWT para Carteira:

- ☒ Tokens com expiração controlada
- ☒ Renovação automática
- ☒ Issuer customizável
- ☒ Secret configurável

### Variáveis de Ambiente:

- `WALLET_JWT_SECRET` - Chave secreta para JWT
  - `WALLET_ISSUER` - Emissor dos tokens
  - `SETUP_TOKEN` - Token para endpoints protegidos
- 



## 12. Integrações Opcionais

### E-mail (SMTP):

- Configuração para envio de emails
- Variáveis: `SMTP_HOST`, `SMTP_USER`, `SMTP_PASS`

### WhatsApp Cloud API:

- Integração para envio de mensagens
- Variáveis: `WHATSAPP_TOKEN`, `WHATSAPP_FROM_PHONE_ID`

### OpenAI:

- Integração para funcionalidades de IA
  - Variável: `OPENAI_API_KEY`
-

## 13. Estrutura Técnica

### Dependencies Principais:

- Next.js 14.2.4
- React 18.2.0
- Supabase JS 2.45.0
- MercadoPago 2.1.8
- html5-qrcode 2.3.8
- qrcode 1.5.3
- jose 5.2.3 (JWT)
- nodemailer 6.9.13
- OpenAI 4.60.0

### Estrutura de Pastas:

```
app/  
├── (auth)/login/      # Autenticação  
├── dashboard/         # Dashboard do usuário  
├── wallet/            # Carteira digital  
├── academias/scanner/ # Scanner de QR  
├── admin/             # Painel admin  
├── tickets/           # Sistema de suporte  
├── crm/leads/         # Gestão de leads  
└── api/              # APIs backend
```

## Fluxo do Usuário Típico:

1. **Acesso inicial** → Página home com opções de login
2. **Login** → Magic link via email
3. **Dashboard** → Visualiza status do plano e pagamentos
4. **Pagamento** → Processa via MercadoPago se necessário
5. **Carteira** → Acessa QR Code dinâmico

6. **Check-in** → Scanner valida entrada na academia

7. **Suporte** → Sistema de tickets para dúvidas

---

**Total de Funcionalidades Identificadas:** 13 módulos principais com mais de 50 funcionalidades específicas implementadas.