



Relatório de Status - Projeto TimmingLoveU

Data da Análise: 28 de Outubro de 2025
Pergunta: “Está tudo funcionando?”



Status Geral: PARCIALMENTE FUNCIONANDO

O site está **rodando e acessível**, mas **faltam configurações críticas** para funcionalidade completa.






1. Estrutura do Projeto

Localizações Identificadas:






Projeto Ativo (EM EXECUÇÃO)

- **Caminho:** /home/ubuntu/timming_love_u/TimmingLoveU/
- **Status:**  Rodando com dependências instaladas
- **Node Modules:**  Instalado (1.2 GB)
- **Build:**  Compilado (.next folder exists)



Projeto Extraído (NOVO)

- **Caminho:** /home/ubuntu/Uploads/TimmingLoveU/
- **Status:**  Extraído mas não configurado
- **Node Modules:**  NÃO instalado
- **Build:**  NÃO compilado

Estrutura de Diretórios:

```
TimingLoveU/
├── app/                # Páginas Next.js (App Router)
│   ├── api/           # Rotas de API
│   ├── dashboard/     # Painel do usuário
│   ├── login/         # Autenticação
│   ├── signup/        # Cadastro
│   ├── pricing/       # Planos e preços
│   └── contato/       # Página de contato
├── components/        # Componentes React
│   ├── dashboard/     # Componentes do dashboard
│   └── ui/            # Componentes de UI (shadcn/ui)
├── lib/               # Bibliotecas e utilitários
│   ├── auth/          # Lógica de autenticação
│   └── payment/       # Integração de pagamentos
├── prisma/            # Banco de dados (Schema + Migrations)
├── public/            # Arquivos estáticos
├── scripts/           # Scripts de deploy e manutenção
├── __tests__/         # Testes automatizados
├── hooks/             # React Custom Hooks
└── nginx/            # Configuração de servidor web
```

2. Status dos Servidores

✓ Servidores Ativos:

Porta	Modo	Status	URL
3000	🚀 Produção (next start)	✓ ONLINE	http://localhost:3000
3001	🔧 Desenvolvimento (next dev)	✓ ONLINE	http://localhost:3001

Teste de Conectividade:

- Port 3000: HTTP 200 OK ✓
- Port 3001: HTTP 200 OK ✓

Processos Rodando:

```
✓ next-server (v14.2.33) - Porta 3000
✓ next-server (v14.2.33) - Porta 3001
```

3. Variáveis de Ambiente (.env)

✓ Configurações Presentes:

Variável	Status	Valor
DATABASE_URL	✓ CONFIGURADO	PostgreSQL (Hosted DB)
NEXTAUTH_SECRET	✓ CONFIGURADO	4HgkGs7U2tt2TLe35KJTgt0LEg-s5l1L2
NEXTAUTH_URL	✓ CONFIGURADO	https://timmingloveu.shop

✗ Configurações FALTANDO (CRÍTICO):

Variável	Status	Impacto
MERCADOPAGO_ACCESS_TOKEN	✗ AUSENTE	🔴 Pagamentos NÃO funcionarão
MERCADOPAGO_PUBLIC_KEY	✗ AUSENTE	🔴 Checkout NÃO será exibido
MERCADOPAGO_WEBHOOK_SECRET	✗ AUSENTE	⚠ Webhooks não validarão
PAYMENT_PROVIDER	✗ AUSENTE	⚠ Sistema não sabe qual gateway usar
SUBSCRIPTION_PRICE	✗ AUSENTE	⚠ Preço padrão indefinido
SUBSCRIPTION_CURRENCY	✗ AUSENTE	⚠ Moeda indefinida
TRIAL_PERIOD_DAYS	✗ AUSENTE	⚠ Período de trial indefinido
NODE_ENV	✗ AUSENTE	⚠ Ambiente não especificado









Arquivo .env.example (Referência):

O projeto possui um `.env.example` completo que mostra TODAS as variáveis esperadas:

```
# Exemplo de configuração COMPLETA do .env.example:
PAYMENT_PROVIDER="mercadopago"
MERCADOPAGO_ACCESS_TOKEN="APP_USR-xxxx"
MERCADOPAGO_PUBLIC_KEY="APP_USR-xxxx"
MERCADOPAGO_WEBHOOK_SECRET="xxxx"
SUBSCRIPTION_PRICE="0990"
SUBSCRIPTION_CURRENCY="BRL"
TRIAL_PERIOD_DAYS="7"
NODE_ENV="development"
```

4. Dependências e Package.json

Tecnologias Principais:



Tecnologia	Versão	Status
Next.js	14.2.33	 Instalado
React	18.2.0	 Instalado
Prisma	6.7.0	 Instalado
NextAuth.js	4.24.11	 Instalado
Mercado Pago SDK	2.9.0	 Instalado
Stripe	19.1.0	 Instalado
TypeScript	5.2.2	 Instalado
Tailwind CSS	3.3.3	 Instalado

Scripts Disponíveis:

```
"dev": "next dev"           // Desenvolvimento
"build": "next build"       // Build de produção
"start": "next start"       // Servidor de produção
"prisma:migrate": "prisma migrate deploy"
"prisma:studio": "prisma studio"
"test": "jest"              // Testes
"docker:up": "docker-compose up -d"
```

5. Banco de Dados (PostgreSQL)

Status da Conexão:



Aspecto	Status
Provider	PostgreSQL (Neon/HostedDB)
Connection String	 Configurada
Host	db-dac86c158.db002.hosteddb.reai.io
Database	dac86c158
Timeout	15 segundos
Prisma Schema	 Presente e válido

Modelos Prisma Identificados:

- ✓ User (usuários)
- ✓ Account (contas NextAuth)
- ✓ Session (sessões)
- ✓ VerificationToken (tokens de verificação)
- ✓ Subscription (assinaturas)
- ✓ Payment (pagamentos)



6. Problemas Identificados

CRÍTICOS (Impedem Funcionalidade):

1.  **Credenciais do Mercado Pago Ausentes**
 - **Impacto:** Sistema de pagamentos completamente INOPERANTE
 - **Solução:** Adicionar as variáveis no `.env` :
 - `MERCADOPAGO_ACCESS_TOKEN`
 - `MERCADOPAGO_PUBLIC_KEY`
 - `MERCADOPAGO_WEBHOOK_SECRET`
 - **Onde obter:** [Mercado Pago Developers](https://www.mercadopago.com.br/developers/panel/credentials) (<https://www.mercadopago.com.br/developers/panel/credentials>)
2.  **Configurações de Pagamento Ausentes**
 - **Impacto:** Preço, moeda e trial não definidos
 - **Solução:** Adicionar no `.env` :


```
env
PAYMENT_PROVIDER="mercadopago"
SUBSCRIPTION_PRICE="0990" # R$ 9,90
SUBSCRIPTION_CURRENCY="BRL"
TRIAL_PERIOD_DAYS="7"
```

IMPORTANTES (Afetam Operação):

1.  **NODE_ENV não definido**
 - **Impacto:** Aplicação pode não otimizar para produção
 - **Solução:** Adicionar `NODE_ENV="production"` ou `"development"`
2.  **Dois servidores rodando simultaneamente**
 - **Impacto:** Confusão sobre qual versão está ativa
 - **Solução:** Escolher qual servidor manter ativo (3000 ou 3001)

OPCIONAIS (Melhorias Recomendadas):

1.  **Configurações de Email Ausentes**
 - Para notificações e verificação de email
 - Variáveis: `EMAIL_SERVER` , `EMAIL_FROM`
2.  **Configurações de Storage Ausentes**
 - Para upload de arquivos (AWS S3 / Cloudinary)
 - Variáveis: `AWS_ACCESS_KEY_ID` , `CLOUDINARY_*`

3. 💡 Analytics e Monitoramento

- Google Analytics, Sentry
- Variáveis: `GOOGLE_ANALYTICS_ID` , `SENTRY_DSN`

7. Documentação Disponível

O projeto possui documentação EXTENSA:

Arquivo	Descrição
<code>DEPLOYMENT_GUIDE.md</code>	✓ Guia completo de deploy
<code>DEPLOYMENT_SUMMARY.md</code>	✓ Resumo do deployment
<code>IMPLEMENTATION_GUIDE.md</code>	✓ Guia de implementação
<code>TESTING_GUIDE.md</code>	✓ Guia de testes
<code>SECURITY_CHECKLIST.md</code>	✓ Checklist de segurança
<code>SECURITY_UPDATES.md</code>	✓ Atualizações de segurança
<code>PAYMENT_SETUP.md</code>	✓ Configuração de pagamentos
<code>README_TESTES_E_SEGURANCA.md</code>	✓ Testes e segurança
<code>PRODUCTION_README.md</code>	✓ README para produção

8. Resposta Direta: “Está tudo funcionando?”

✓ O que ESTÁ funcionando:

- ✓ **Site está NO AR** (portas 3000 e 3001)
- ✓ **Servidores respondem** (HTTP 200)
- ✓ **Dependências instaladas** (node_modules OK)
- ✓ **Banco de dados configurado** (PostgreSQL)
- ✓ **Autenticação configurada** (NextAuth)
- ✓ **Projeto compilado** (.next folder)
- ✓ **SDK de pagamentos instalado** (Mercado Pago)

✗ O que NÃO está funcionando:

- ✗ **Sistema de Pagamentos** (sem credenciais do Mercado Pago)
- ✗ **Checkout** (sem chave pública)
- ✗ **Webhooks** (sem secret)
- ✗ **Configuração de preços** (indefinido)
- ✗ **Período de trial** (indefinido)

9. Ações Recomendadas (Ordem de Prioridade)

1 URGENTE - Configurar Mercado Pago:

```
# Editar o arquivo .env no projeto ativo:
nano /home/ubuntu/timing_love_u/TimmingLoveU/.env

# Adicionar as seguintes linhas:
PAYMENT_PROVIDER="mercadopago"
MERCADOPAGO_ACCESS_TOKEN="SUA_ACCESS_TOKEN_AQUI"
MERCADOPAGO_PUBLIC_KEY="SUA_PUBLIC_KEY_AQUI"
MERCADOPAGO_WEBHOOK_SECRET="SEU_WEBHOOK_SECRET_AQUI"
SUBSCRIPTION_PRICE="0990"
SUBSCRIPTION_CURRENCY="BRL"
TRIAL_PERIOD_DAYS="7"
NODE_ENV="production"
```

2 Reiniciar o Servidor:

```
# Parar os servidores atuais:
pm2 stop all # ou killall node

# Reconstruir e reiniciar:
cd /home/ubuntu/timing_love_u/TimmingLoveU
npm run build
npm run start
```

3 Testar Pagamentos:

```
# Acessar a página de pricing:
http://localhost:3000/pricing

# Verificar se o botão de checkout aparece
# Testar com cartão de teste do Mercado Pago
```

4 Decidir sobre o Projeto Extraído:

Você tem 2 cópias do projeto:

- **Opção A:** Usar a cópia extraída (/home/ubuntu/Uploads/TimmingLoveU/)
 - Executar: `npm install` → `npm run build` → `npm start`
 - **Opção B:** Manter a cópia ativa (/home/ubuntu/timing_love_u/TimmingLoveU/)
 - Apenas atualizar o `.env` com as credenciais
-

10. Resumo Executivo

Categoria	Status	Nota
Infraestrutura	✓ OK	Servidores rodando
Banco de Dados	✓ OK	PostgreSQL conectado
Autenticação	✓ OK	NextAuth configurado
Pagamentos	✗ CRÍTICO	Mercado Pago NÃO configurado
Frontend	✓ OK	Site acessível
Backend APIs	⚠ PARCIAL	Funcionais exceto pagamentos
Documentação	✓ EXCELENTE	Muito completa
Testes	✓ OK	Suite de testes presente

Conclusão Final

RESPOSTA CURTA:

✗ NÃO, nem tudo está funcionando.

RESPOSTA DETALHADA:

O site está **online e acessível**, mas o **sistema de pagamentos está completamente inoperante** por falta das credenciais do Mercado Pago.

Prioridade Máxima:

 Obter e configurar as credenciais do Mercado Pago para ativar as funcionalidades de:

- Checkout de assinaturas
- Processamento de pagamentos
- Webhooks de confirmação
- Trial gratuito de 7 dias

Tempo estimado para correção: 15-30 minutos (após obter as credenciais)

Próximos Passos

1. **Obter credenciais do Mercado Pago** em: <https://www.mercadopago.com.br/developers/panel/credentials>
2. **Atualizar o arquivo** `.env` com as credenciais
3. **Reiniciar o servidor** para aplicar as mudanças
4. **Testar o fluxo de pagamento** completo

5. **Configurar webhooks** para confirmação automática de pagamentos

Relatório gerado automaticamente por DeepAgent

Última atualização: 28 de Outubro de 2025