



Guia Completo de Deploy na Vercel - TimmingLoveU

Data: 28 de Outubro de 2025

Projeto: TimmingLoveU

Domínio: <https://timmingloveu.shop>

Plataforma: Vercel



Índice

1. [Pré-requisitos](#)
 2. [Preparação do Projeto](#)
 3. [Deploy Inicial na Vercel](#)
 4. [Configuração de Variáveis de Ambiente](#)
 5. [Configuração do Domínio Customizado](#)
 6. [Configuração do Webhook do Mercado Pago](#)
 7. [Verificação e Testes](#)
 8. [Troubleshooting](#)
 9. [Manutenção e Atualizações](#)
-



Pré-requisitos

Antes de iniciar o deploy, certifique-se de ter:

- ☒ Conta na [Vercel](https://vercel.com) (<https://vercel.com>) (gratuita ou paga)
 - ☒ Repositório Git (GitHub, GitLab ou Bitbucket) com o código do projeto
 - ☒ Credenciais do Mercado Pago (Public Key, Access Token, Webhook Secret)
 - ☒ Banco de dados PostgreSQL configurado e acessível
 - ☒ Domínio customizado (timmingloveu.shop) com acesso ao painel DNS
 - ☒ Node.js 18+ instalado localmente (para testes)
-



Preparação do Projeto

1. Verificar Estrutura do Projeto

Certifique-se de que o projeto possui os seguintes arquivos:

```

TimingLoveU/
└─ app/           # Código da aplicação Next.js
└─ components/   # Componentes React
└─ lib/          # Bibliotecas e utilitários
└─ prisma/       # Schema do banco de dados
└─ public/       # Arquivos estáticos
└─ .env.production # Variáveis de ambiente (NÃO commitar!)
└─ .env.example  # Template de variáveis
└─ .gitignore    # Arquivos ignorados pelo Git
└─ next.config.js # Configuração do Next.js
└─ package.json  # Dependências do projeto
└─ vercel.json   # Configuração da Vercel
└─ README.md     # Documentação

```

2. Verificar .gitignore

Certifique-se de que o arquivo `.gitignore` contém:

```

# Environment variables
.env
.env.local
.env.production
.env.development

# Next.js
.next/
out/
build/
dist/

# Dependencies
node_modules/

# Logs
logs/
*.log

# Database
*.db
*.sqlite

# OS
.DS_Store
Thumbs.db

```

3. Commitar e Enviar para o Git

```

# Adicionar arquivos ao Git (exceto .env.production)
git add .

# Commitar mudanças
git commit -m "Preparar projeto para deploy na Vercel"

# Enviar para o repositório remoto
git push origin main

```

 **IMPORTANTE:** NUNCA commite o arquivo `.env.production` com valores reais!

Deploy Inicial na Vercel

Passo 1: Importar Projeto

1. Acesse vercel.com (<https://vercel.com>) e faça login
2. Clique em “Add New...” → “Project”
3. Selecione o repositório Git do TimmingLoveU
4. Clique em “Import”

Passo 2: Configurar Build Settings

Na tela de configuração do projeto:

- **Framework Preset:** Next.js (detectado automaticamente)
- **Root Directory:** `./` (raiz do projeto)
- **Build Command:** `npm run build` (padrão)
- **Output Directory:** `.next` (padrão)
- **Install Command:** `npm install` (padrão)

Passo 3: NÃO Deploy Ainda!

 **IMPORTANTE:** Clique em “Environment Variables” ANTES de fazer o deploy inicial.

Configuração de Variáveis de Ambiente

Lista Completa de Variáveis Obrigatórias

Configure TODAS as variáveis abaixo no painel da Vercel:

1. Database Configuration

Variável	Valor	Ambiente
<code>DATABASE_URL</code>	<code>postgresql:// role_dac86c158:m_dnWSpS3tJG wu1CihK2xpxceTzPse_W@db- dac86c158.db002.hosteddb.re ai.io:5432/dac86c158?con- nect_timeout=15</code>	Production

2. NextAuth Configuration

Variável	Valor	Ambiente
<code>NEXTAUTH_SECRET</code>	<code>4HgkGs7U2tt2TLe35KJTGt0LEG- s5l1L2</code>	Production
<code>NEXTAUTH_URL</code>	<code>https://timmingloveu.shop</code>	Production

3. Mercado Pago Configuration

Variável	Valor	Ambiente
PAYMENT_PROVIDER	mercadopago	Production
MERCADOPAGO_PUBLIC_KEY	APP_USR-9cf7e4b9-b410-4d9e-b8f3-6f9e29ce2a0c	Production
MERCADOPAGO_ACCESS_TOKEN	APP_USR-1678121204668263-102301-c2a9d6bed0f5c2648c3811549d6e1309-712155332	Production
MERCADOPAGO_WEBHOOK_SECRET	9YxWOXIRuke1UaTzJyDsigTD-FNU0XSjA0c0+5aUTh3c=	Production

4. Subscription Configuration

Variável	Valor	Ambiente
SUBSCRIPTION_PRICE	990	Production
SUBSCRIPTION_CURRENCY	BRL	Production
TRIAL_PERIOD_DAYS	7	Production

5. Application Configuration

Variável	Valor	Ambiente
NODE_ENV	production	Production
PORT	3000	Production

Como Adicionar as Variáveis

1. Na tela de configuração do projeto, clique em **“Environment Variables”**
2. Para cada variável:
 - **Key:** Nome da variável (ex: DATABASE_URL)
 - **Value:** Valor da variável (copie exatamente como mostrado acima)
 - **Environment:** Selecione **“Production”**
3. Clique em **“Add”** para cada variável
4. Repita para todas as 14 variáveis obrigatórias

Método Alternativo: Adicionar Após Deploy

Se você já fez o deploy, pode adicionar as variáveis depois:

1. Acesse o dashboard do projeto na Vercel
2. Vá em **Settings** → **Environment Variables**

3. Adicione cada variável conforme instruções acima
4. Após adicionar todas, clique em **“Deployments”**
5. No último deployment, clique nos três pontos “...” → **“Redeploy”**
6. Marque **“Use existing Build Cache”** (opcional, para deploy mais rápido)
7. Clique em **“Redeploy”**



Configuração do Domínio Customizado

Passo 1: Adicionar Domínio na Vercel

1. No dashboard do projeto, vá em **Settings → Domains**
2. Clique em **“Add”**
3. Digite: `timmingloveu.shop`
4. Clique em **“Add”**

Passo 2: Configurar DNS

A Vercel fornecerá instruções específicas. Geralmente você precisa adicionar:

Opção A: Usar Nameservers da Vercel (Recomendado)

Se você tem acesso total ao domínio:

1. Vá ao painel do seu registrador de domínio
2. Altere os nameservers para:
`ns1.vercel-dns.com`
`ns2.vercel-dns.com`
3. Aguarde propagação (pode levar até 48h, geralmente 1-2h)

Opção B: Adicionar Registros DNS Manualmente

Se você não pode alterar os nameservers:


1. Adicione um registro **A** apontando para o IP da Vercel:

Type: A
Name: @
Value: 76.76.21.21
TTL: 3600

2. Adicione um registro **CNAME** para `www`:

Type: CNAME
Name: www
Value: cname.vercel-dns.com
TTL: 3600

Passo 3: Verificar Configuração

1. Aguarde alguns minutos para propagação
2. A Vercel verificará automaticamente a configuração
3. Quando configurado corretamente, você verá um  verde
4. O SSL será provisionado automaticamente (pode levar até 24h)

Passo 4: Atualizar NEXTAUTH_URL

⚠ **IMPORTANTE:** Após configurar o domínio, verifique se a variável `NEXTAUTH_URL` está correta:

1. Vá em **Settings** → **Environment Variables**
2. Verifique se `NEXTAUTH_URL = https://timingloveu.shop`
3. Se precisar alterar, edite e faça um **Redeploy**



Configuração do Webhook do Mercado Pago

O webhook é essencial para receber notificações de pagamentos em tempo real.

Passo 1: Acessar Painel do Mercado Pago

1. Acesse <https://www.mercadopago.com.br/developers/panel/webhooks> (<https://www.mercadopago.com.br/developers/panel/webhooks>)
2. Faça login com sua conta do Mercado Pago
3. Selecione a aplicação correta

Passo 2: Configurar Webhook

1. Clique em **“Configurar notificações”** ou **“Criar webhook”**
2. Preencha os campos:
 - **Nome:** TimmingLoveU Production Webhook
 - **URL de notificação:** `https://timingloveu.shop/api/webhooks/mercadopago`
 - **Eventos:** Selecione todos os eventos de pagamento:
 - ☒ `payment.created`
 - ☒ `payment.updated`
 - ☒ `subscription.created`
 - ☒ `subscription.updated`
 - ☒ `subscription.cancelled`
 - ☒ `subscription.paused`
 - ☒ `subscription.authorized`
 - ☒ `subscription.preapproval_plan.updated`

1. Clique em **“Salvar”** ou **“Criar”**

Passo 3: Obter Webhook Secret

1. Após criar o webhook, o Mercado Pago exibirá um **Webhook Secret**
2. Copie este valor
3. Verifique se corresponde ao valor em `MERCADOPAGO_WEBHOOK_SECRET` na Vercel
4. Se for diferente, atualize a variável na Vercel e faça um Redeploy

Passo 4: Testar Webhook

1. No painel do Mercado Pago, procure por **“Testar webhook”** ou **“Enviar teste”**
2. Clique para enviar uma notificação de teste
3. Verifique os logs na Vercel:
 - Vá em **Deployments** → Clique no deployment ativo

- Clique em **“Functions”** → Selecione a função do webhook
- Verifique se a requisição foi recebida com sucesso (status 200)

Estrutura do Endpoint do Webhook

O endpoint do webhook deve estar em:

```
app/api/webhooks/mercadopago/route.ts
```

Exemplo de implementação:

```

import { NextRequest, NextResponse } from 'next/server';
import crypto from 'crypto';

export async function POST(request: NextRequest) {
  try {
    // Obter o corpo da requisição
    const body = await request.text();

    // Obter o header de assinatura
    const signature = request.headers.get('x-signature');
    const requestId = request.headers.get('x-request-id');

    // Validar assinatura
    const webhookSecret = process.env.MERCADOPAGO_WEBHOOK_SECRET!;
    const expectedSignature = crypto
      .createHmac('sha256', webhookSecret)
      .update(body)
      .digest('hex');

    if (signature !== expectedSignature) {
      return NextResponse.json(
        { error: 'Invalid signature' },
        { status: 401 }
      );
    }

    // Processar notificação
    const data = JSON.parse(body);

    // Lógica de processamento baseada no tipo de evento
    switch (data.type) {
      case 'payment':
        await handlePaymentNotification(data);
        break;
      case 'subscription':
        await handleSubscriptionNotification(data);
        break;
      default:
        console.log('Tipo de notificação não tratado:', data.type);
    }

    return NextResponse.json({ success: true }, { status: 200 });
  } catch (error) {
    console.error('Erro ao processar webhook:', error);
    return NextResponse.json(
      { error: 'Internal server error' },
      { status: 500 }
    );
  }
}




```

✓ Verificação e Testes

1. Verificar Deploy

Após o deploy, verifique:

- ✓ Build completou com sucesso (sem erros)

-  Aplicação está acessível em `https://timmingloveu.shop`
-  SSL está ativo (cadeado verde no navegador)
-  Página inicial carrega corretamente

2. Testar Funcionalidades Principais

Teste de Autenticação

1. Acesse `https://timmingloveu.shop/login`
2. Tente fazer login com credenciais válidas
3. Verifique se o login funciona corretamente
4. Teste o logout

Teste de Registro

1. Acesse `https://timmingloveu.shop/register`
2. Crie uma nova conta
3. Verifique se o registro funciona
4. Confirme que o usuário foi criado no banco de dados

Teste de Conexão com Banco de Dados

1. Verifique os logs da Vercel:
 - Vá em **Deployments** → Clique no deployment ativo
 - Clique em **“Functions”**
 - Procure por logs de conexão com o banco
2. Não deve haver erros de conexão

Teste de Pagamento (Ambiente de Teste)

 **IMPORTANTE:** Use credenciais de teste do Mercado Pago primeiro!

1. Acesse a página de assinatura
2. Clique em “Assinar”
3. Use um cartão de teste do Mercado Pago:
 - **Cartão:** 5031 4332 1540 6351
 - **CVV:** 123
 - **Validade:** 11/25
 - **Nome:** APRO (para aprovar)
4. Complete o pagamento
5. Verifique se o webhook foi recebido nos logs
6. Confirme que a assinatura foi criada no banco

3. Verificar Logs

Monitore os logs para identificar possíveis erros:

1. Vá em **Deployments** → Clique no deployment ativo
2. Clique em **“Functions”**
3. Selecione diferentes funções para ver seus logs
4. Procure por erros (linhas em vermelho)

4. Testar Performance

Use ferramentas para verificar a performance:

- [PageSpeed Insights](https://pagespeed.web.dev/) (<https://pagespeed.web.dev/>)

- [GTmetrix](https://gtmetrix.com/) (https://gtmetrix.com/)
- [WebPageTest](https://www.webpagetest.org/) (https://www.webpagetest.org/)

Meta: Score acima de 90 no PageSpeed Insights

Troubleshooting

Problema: Build Falha

Sintomas: Deploy falha durante o build

Soluções:

1. Verifique os logs de build na Vercel
2. Procure por erros de TypeScript ou ESLint
3. Teste o build localmente:

```
bash
```

```
npm run build
```

4. Corrija os erros e faça um novo commit

Problema: Erro de Conexão com Banco de Dados

Sintomas: Erro “Can’t reach database server” ou timeout

Soluções:

1. Verifique se `DATABASE_URL` está correta
2. Confirme que o banco de dados está acessível publicamente
3. Verifique se há firewall bloqueando conexões da Vercel
4. Teste a conexão localmente:

```
bash
```

```
npx prisma db pull
```

5. Adicione `?connect_timeout=15` na URL do banco

Problema: NextAuth Não Funciona

Sintomas: Erro ao fazer login, sessão não persiste

Soluções:

1. Verifique se `NEXTAUTH_SECRET` está configurado
2. Confirme que `NEXTAUTH_URL` corresponde ao domínio real
3. Verifique se o domínio tem SSL ativo
4. Limpe cookies do navegador e teste novamente
5. Verifique os logs para erros específicos

Problema: Webhook Não Recebe Notificações

Sintomas: Pagamentos não são processados automaticamente

Soluções:

1. Verifique se a URL do webhook está correta no painel do Mercado Pago
2. Confirme que `MERCADOPAGO_WEBHOOK_SECRET` está correto

3. Teste o webhook manualmente no painel do Mercado Pago
4. Verifique os logs da função do webhook na Vercel
5. Certifique-se de que o endpoint retorna status 200
6. Verifique se a validação de assinatura está correta

Problema: Domínio Não Resolve

Sintomas: Site não acessível pelo domínio customizado

Soluções:

1. Verifique a configuração DNS:

```
bash
nslookup timingloveu.shop
dig timingloveu.shop
```

2. Aguarde propagação DNS (pode levar até 48h)

3. Limpe o cache DNS local:

```
```bash
Windows
ipconfig /flushdns
```

# macOS

```
sudo dscacheutil -flushcache
```

# Linux

```
sudo systemd-resolve --flush-caches
```

```
```
```

4. Verifique se os nameservers ou registros A/CNAME estão corretos
5. Use [DNS Checker](https://dnschecker.org/) (<https://dnschecker.org/>) para verificar propagação global

Problema: SSL Não Ativa

Sintomas: Navegador mostra “Não seguro” ou erro de certificado

Soluções:

1. Aguarde até 24h para provisionamento do SSL
2. Verifique se o domínio está corretamente configurado na Vercel
3. Remova e adicione o domínio novamente
4. Verifique se há registros CAA no DNS bloqueando Let’s Encrypt
5. Entre em contato com o suporte da Vercel se persistir

Problema: Variáveis de Ambiente Não Funcionam

Sintomas: Aplicação não consegue acessar variáveis de ambiente

Soluções:

1. Verifique se as variáveis estão configuradas para “Production”
2. Faça um **Redeploy** após adicionar/alterar variáveis
3. Não use `NEXT_PUBLIC_` para variáveis sensíveis (elas são expostas no frontend)
4. Verifique se os nomes das variáveis estão corretos (case-sensitive)
5. Teste localmente com `.env.local` para confirmar que o código está correto

Problema: Pagamentos Não Processam

Sintomas: Erro ao tentar fazer pagamento

Soluções:

1. Verifique se `MERCADOPAGO_ACCESS_TOKEN` está correto
2. Confirme que está usando credenciais de produção (não teste)
3. Verifique se a conta do Mercado Pago está ativa e verificada
4. Teste com cartão de teste primeiro
5. Verifique os logs da API do Mercado Pago
6. Confirme que `SUBSCRIPTION_PRICE` está em centavos (990 = R\$ 9,90)

Manutenção e Atualizações

Deploy de Novas Versões

1. Faça as alterações no código localmente
2. Teste localmente:

```
bash
npm run dev
npm run build
```
3. Commit e push para o Git:

```
bash
git add .
git commit -m "Descrição das mudanças"
git push origin main
```
4. A Vercel fará deploy automaticamente
5. Verifique o deploy no dashboard da Vercel
6. Teste a nova versão em produção

Rollback para Versão Anterior

Se algo der errado:

1. Vá em **Deployments** na Vercel
2. Encontre o deployment anterior que funcionava
3. Clique nos três pontos **"..."** → **"Promote to Production"**
4. Confirme a ação
5. O site voltará para a versão anterior imediatamente

Monitoramento Contínuo

Configure alertas para monitorar a saúde da aplicação:

1. **Vercel Analytics:**
 - Ative em **Settings** → **Analytics**
 - Monitore performance e erros
2. **Uptime Monitoring:**
 - Use serviços como [UptimeRobot](https://uptimerobot.com/) (<https://uptimerobot.com/>)

- Configure para verificar `https://timmingloveu.shop` a cada 5 minutos
- Receba alertas por email se o site ficar fora do ar

3. Error Tracking:

- Integre com [Sentry](https://sentry.io/) (<https://sentry.io/>)
- Adicione `SENTRY_DSN` nas variáveis de ambiente
- Receba notificações de erros em tempo real

Backup do Banco de Dados

Configure backups regulares:

1. Use a ferramenta de backup do seu provedor de banco de dados
2. Configure backups automáticos diários
3. Teste a restauração periodicamente
4. Mantenha backups por pelo menos 30 dias

Atualização de Dependências

Mantenha as dependências atualizadas:

```
# Verificar dependências desatualizadas
npm outdated

# Atualizar dependências menores
npm update

# Atualizar dependências maiores (com cuidado)
npm install package@latest

# Verificar vulnerabilidades
npm audit

# Corrigir vulnerabilidades automaticamente
npm audit fix
```

Rotação de Secrets

Por segurança, rotacione secrets periodicamente (a cada 3-6 meses):

1. NEXTAUTH_SECRET:

```
bash
openssl rand -base64 32
```

- Atualize na Vercel
- Faça Redeploy
- Usuários precisarão fazer login novamente

2. MERCADOPAGO_WEBHOOK_SECRET:

- Gere novo secret no painel do Mercado Pago
- Atualize na Vercel
- Faça Redeploy

3. DATABASE_URL:

- Rotacione senha do banco de dados
- Atualize a URL na Vercel
- Faça Redeploy imediatamente

Suporte e Recursos

Documentação Oficial

- [Vercel Documentation](https://vercel.com/docs) (https://vercel.com/docs)
- [Next.js Documentation](https://nextjs.org/docs) (https://nextjs.org/docs)
- [Mercado Pago Developers](https://www.mercadopago.com.br/developers) (https://www.mercadopago.com.br/developers)
- [Prisma Documentation](https://www.prisma.io/docs) (https://www.prisma.io/docs)

Comunidade

- [Vercel Discord](https://vercel.com/discord) (https://vercel.com/discord)
- [Next.js GitHub Discussions](https://github.com/vercel/next.js/discussions) (https://github.com/vercel/next.js/discussions)
- [Stack Overflow](https://stackoverflow.com/questions/tagged/vercel) (https://stackoverflow.com/questions/tagged/vercel)

Suporte Direto

- **Vercel Support:** vercel.com/support (https://vercel.com/support)
- **Mercado Pago Support:** www.mercadopago.com.br/developers/support (https://www.mercadopago.com.br/developers/support)

Checklist Final







Antes de considerar o deploy completo, verifique:

- ☐ Projeto commitado no Git (sem .env.production)
- ☐ Deploy inicial na Vercel realizado
- ☐ Todas as 14 variáveis de ambiente configuradas
- ☐ Domínio customizado configurado e SSL ativo
- ☐ Webhook do Mercado Pago configurado e testado
- ☐ Login/Registro funcionando
- ☐ Conexão com banco de dados funcionando
- ☐ Pagamento de teste realizado com sucesso
- ☐ Logs verificados (sem erros críticos)
- ☐ Performance testada (score > 90)
- ☐ Monitoramento configurado (Uptime, Analytics)
- ☐ Backup do banco de dados configurado
- ☐ Documentação atualizada
- ☐ Equipe treinada para manutenção

Conclusão

Parabéns! Se você seguiu todos os passos, o TimmingLoveU está agora rodando em produção na Vercel com:

- ☒ Deploy automatizado via Git

-  Domínio customizado com SSL
-  Banco de dados PostgreSQL conectado
-  Autenticação NextAuth funcionando
-  Pagamentos via Mercado Pago integrados
-  Webhooks configurados para notificações em tempo real
-  Monitoramento e logs ativos

Próximos Passos:

1. Monitore os primeiros usuários e pagamentos
2. Ajuste conforme feedback
3. Implemente melhorias incrementais
4. Mantenha as dependências atualizadas
5. Faça backups regulares

Lembre-se:

- Monitore os logs regularmente
- Responda rapidamente a erros
- Mantenha a documentação atualizada
- Faça testes antes de cada deploy
- Comunique mudanças à equipe

Documento criado em: 28 de Outubro de 2025

Versão: 1.0

Autor: DeepAgent - Abacus.AI

Projeto: TimmingLoveU

Para dúvidas ou suporte adicional, consulte a documentação oficial ou entre em contato com a equipe de desenvolvimento.