

# DIAGNÓSTICO E CORREÇÃO - CHECKOUT ASAAS/EFI

## RESUMO DO PROBLEMA





Data: 27/11/2025

Status:  Correções Aplicadas + Logs de Debug Adicionados

### Sintoma Reportado:

- Após pagamento aprovado no Asaas, a tela de checkout permanece com status `pending`
- A rota `/api/checkout/check-payment` retorna sempre `gatewayStatus: null`
- O sistema não libera o acesso automaticamente

### Causa Prováveis Identificadas:

1.  Erro silencioso na consulta ao gateway (capturado mas não detalhado)
2.  `stripeSessionId` inválido ou ausente
3.  Configuração de ambiente incorreta (sandbox vs production)
4.  Credenciais do Asaas inválidas ou expiradas

## CORREÇÕES APLICADAS

### 1. Logs Detalhados Adicionados

Arquivo: `app/api/checkout/check-payment/route.ts`

#### O que foi feito:

-  Log inicial completo dos dados do pagamento:

typescript

```
console.log("💰 [Check Payment] Dados do pagamento:", {  
  paymentId, gateway, stripeSessionId, currentStatus, amount, createdAt  
});
```

-  Log antes da consulta ao gateway:

typescript

```
console.log("🔍 [Check Payment] Iniciando consulta ao gateway:", {  
  gateway, externalId: stripeSessionId  
});
```

-  Log detalhado da resposta do Asaas:

typescript

```
console.log("📦 [Check Payment] Resposta completa do Asaas:", {  
  id, status, value, billingType, dateCreated, dueDate, invoiceUrl  
});
```

-  Log detalhado de erros:

typescript

```
console.error("❌ [Check Payment] Erro COMPLETO ao consultar gateway:", {
```

```
gateway, externalId, errorName, errorMessage, errorStack, errorResponse
});
```

-  **Log de status final com debug:**

```
typescript
console.log("✅ [Check Payment] Status final:", {
  currentStatus, gatewayStatus, errorMessage, hasStripeSessionId
});
```

## 2. Resposta JSON Enriquecida

**Antes:**

```
{
  "status": "pending",
  "gatewayStatus": null,
  "paymentId": "xxx",
  "amount": 97,
  "gateway": "asaas",
  "planName": "Básico"
}
```

**Agora:**

```
{
  "status": "pending",
  "gatewayStatus": null,
  "paymentId": "xxx",
  "amount": 97,
  "gateway": "asaas",
  "planName": "Básico",
  "errorMessage": "Mensagem de erro se houver",
  "debug": {
    "stripeSessionId": "pay_xxx",
    "gateway": "asaas",
    "currentStatus": "pending"
  }
}
```

## 3. Logs Já Existentes no lib/asaas.ts

A função `asaasRequest` já possui logs detalhados:

```
console.log(`[Asaas Request] GET https://sandbox.asaas.com/api/v3/payments/xxx`);
console.log(`[Asaas Request] Environment: sandbox`);
console.log(`[Asaas Response] Status: 200`);
console.log(`[Asaas Response] Data: { ... }`);
```

---

## COMO DIAGNOSTICAR O PROBLEMA

---

### Passo 1: Acessar os Logs do Servidor

```
cd /home/ubuntu/clivus_landing_page/nextjs_space  
pm2 logs clivus --lines 100 --nostream
```

### Passo 2: Fazer um Pagamento de Teste

1. **Acessar o checkout:**

```
https://clivus.marcosleandru.com.br/checkout?plan=basico
```

2. **Fazer login** com qualquer usuário de teste

3. **Clicar em “Pagar com Boleto ou Cartão”**

4. **Completar o pagamento no Asaas**

5. **Clicar em “Já fiz o pagamento”** na tela de checkout

### Passo 3: Analisar os Logs

**Logs Esperados (Sucesso):**

```

[Check Payment] Dados do pagamento: {
  paymentId: 'cmigunmg80003nr08yziqwhgg',
  gateway: 'asaas',
  stripeSessionId: 'pay_xxxxxxxx', ID do pagamento no Asaas
  currentStatus: 'pending',
  amount: 97
}

[Check Payment] Iniciando consulta ao gateway: {
  gateway: 'asaas',
  externalId: 'pay_xxxxxxxx'
}

[Check Payment] Chamando API Asaas com ID: pay_xxxxxxxx

[Asaas Request] GET https://sandbox.asaas.com/api/v3/payments/pay_xxxxxxxx
[Asaas Request] Environment: sandbox Verifica se está usando o ambiente correto

[Asaas Response] Status: 200
[Asaas Response] Data: {
  "id": "pay_xxxxxxxx",
  "status": "CONFIRMED", Status real no Asaas
  "value": 97,
  "billingType": "CREDIT_CARD",
  ...
}

[Check Payment] Resposta completa do Asaas: {
  id: 'pay_xxxxxxxx',
  status: 'CONFIRMED',
  value: 97,
  billingType: 'CREDIT_CARD',
  invoiceUrl: 'https://sandbox.asaas.com/...'
}

[Check Payment] Status Asaas: {
  original: 'CONFIRMED',
  mapped: 'completed',
  wouldUpdate: true
}

[Check Payment] Status atualizado no banco de PENDING para: completed
[Check Payment] Pagamento confirmado! Liberando acesso...
[Check Payment] Acesso liberado para usuário: xxx
[Check Payment] Senha temporária gerada para usuário
[Check Payment] Emails enviados com sucesso

[Check Payment] Status final: {
  currentStatus: 'completed',
  gatewayStatus: 'CONFIRMED',
  errorMessage: null,
  hasStripeSessionId: true
}

```

### Logs Esperados (Erro):

```

[Check Payment] Dados do pagamento: {
  paymentId: 'cmigunmg80003nr08yziqwhgg',
  gateway: 'asaas',
  stripeSessionId: 'pay_XXXXXXXX',
  currentStatus: 'pending',
  amount: 97
}

[Check Payment] Iniciando consulta ao gateway: {
  gateway: 'asaas',
  externalId: 'pay_XXXXXXXX'
}

[Check Payment] Chamando API Asaas com ID: pay_XXXXXXXX

[Asaas Request] GET https://api.asaas.com/v3/payments/pay_XXXXXXXX ← Produção
[Asaas Request] Environment: production

[Asaas Response] Status: 404 [ERRO: Pagamento não encontrado]
[Asaas Response] Data: {
  "errors": [{
    "code": "invalid_action",
    "description": "Registro não encontrado"
  }]
}

[ERRO] Erro na API Asaas: { errors: [...] }
[ERRO] Mensagem de erro: Registro não encontrado

[ERRO] [Check Payment] Erro COMPLETO ao consultar gateway: {
  gateway: 'asaas',
  externalId: 'pay_XXXXXXXX',
  errorName: 'Error',
  errorMessage: 'Registro não encontrado',
  errorStack: [...],
  errorResponse: 'N/A'
}

[OK] [Check Payment] Status final: {
  currentStatus: 'pending',
  gatewayStatus: null,
  errorMessage: 'Registro não encontrado',
  hasStripeSessionId: true
}

```

## ! POSSÍVEIS PROBLEMAS E SOLUÇÕES

### 1. Pagamento Não Encontrado (404)

Sintoma nos logs:

```

[Asaas Response] Status: 404
"description": "Registro não encontrado"

```

Causas:

- **X** O `stripeSessionId` está errado (não é o ID do Asaas)

- ❌ O gateway está configurado para `production`, mas o pagamento foi feito em `sandbox` (ou vice-versa)

#### Solução:

1. Verificar o `stripeSessionId` no banco de dados:

```
sql
SELECT id, gateway, "stripeSessionId", status, amount
FROM "Payment"
WHERE id = 'cmigunmg80003nr08yziqwhgg';
```

1. Verificar a configuração do gateway Asaas no admin:

<https://clivus.marcosleandru.com.br/admin/gateways>

- Conferir se `Environment` está como `sandbox` ou `production`
- Conferir se as credenciais estão corretas para o ambiente escolhido

2. Verificar o link gerado no checkout:

- O link do Asaas deve começar com:

- **Sandbox:** <https://sandbox.asaas.com/...>

- **Produção:** <https://www.asaas.com/...>

## 2. Credenciais Inválidas (401)

#### Sintoma nos logs:

```
[Asaas Response] Status: 401
"description": "Invalid access token"
```

#### Solução:

1. Verificar a API Key do Asaas no admin:

<https://clivus.marcosleandru.com.br/admin/gateways>

1. Gerar uma nova API Key no painel do Asaas:

- **Sandbox:** <https://sandbox.asaas.com/configuracoes/integracoes>
- **Produção:** <https://www.asaas.com/configuracoes/integracoes>

2. Atualizar as credenciais no admin e testar novamente

## 3. stripeSessionId Nulo ou Ausente

#### Sintoma nos logs:

```
⚠️ [Check Payment] stripeSessionId não encontrado no pagamento
```

#### Solução:

1. Verificar se o checkout está salvando o `stripeSessionId` corretamente

2. Checar o código em `/api/checkout/route.ts` e `/api/checkout/pix/route.ts`

3. O `stripeSessionId` deve ser salvo após criar a cobrança no Asaas:

```
typescript
await prisma.payment.update({
  where: { id: paymentId },
  data: { stripeSessionId: asaasPaymentLink.id }
});
```

## 4. Gateway Não Suportado

### Sintoma nos logs:

 [Check Payment] Gateway não suportado para consulta: stripe

### Solução:

- Atualmente, apenas `asaas` e `efi` são suportados para consulta em tempo real
- Para adicionar suporte a outros gateways (Stripe, CORA, Pagar.me), é necessário:
  1. Criar uma função `getStripePayment` em `lib/stripe.ts`
  2. Adicionar um `else if` em `/api/checkout/check-payment/route.ts`



## CHECKLIST DE DIAGNÓSTICO

### Configuração

- ☐ Gateway Asaas está habilitado em `/admin/gateways`
- ☐ Ambiente (sandbox/production) está correto
- ☐ Credenciais (API Key) estão corretas para o ambiente
- ☐ Link de pagamento gerado está usando o domínio correto

### Pagamento

- ☐ `stripeSessionId` foi salvo no banco após criar a cobrança
- ☐ `stripeSessionId` é um ID válido do Asaas (ex: `pay_xxx` ou `chr_xxx`)
- ☐ Pagamento foi realmente aprovado no painel do Asaas

### Logs

- ☐ Logs mostram a consulta sendo feita ao gateway
- ☐ URL da API está correta (sandbox vs production)
- ☐ Resposta da API Asaas é `200 OK`
- ☐ Status retornado pelo Asaas é `CONFIRMED` ou `RECEIVED`

## TESTES PÓS-CORREÇÃO

### Teste 1: Cartão Sandbox Asaas

```
# 1. Acessar checkout
https://clivus.marcosleandru.com.br/checkout?plan=basico

# 2. Fazer login
Email: teste@teste.com
Senha: 123456

# 3. Pagar com Boleto/Cartão
Cartão de teste: 5162 3068 9088 7704
CVV: 318
Data: qualquer futura

# 4. Aguardar ou clicar em "Já fiz o pagamento"

# 5. Verificar logs
pm2 logs clivus --lines 100 --nostream
```

#### Resultado Esperado:

- ☒ Logs mostram consulta ao Asaas
- ☒ Status retorna `CONFIRMED`
- ☒ Banco é atualizado para `completed`
- ☒ Acesso é liberado ( `hasAccess: true` )
- ☒ Emails são enviados
- ☒ Tela de checkout atualiza para “completed”

### Teste 2: PIX Sandbox Asaas

```
# 1-2. Igual ao Teste 1

# 3. Pagar com PIX
- Copiar código PIX
- Simular pagamento no painel Asaas Sandbox

# 4-5. Igual ao Teste 1
```

## STATUS ATUAL

### Build

- ☒ TypeScript: 0 erros
- ☒ Build: Sucesso
- ☒ 33 páginas geradas
- ☒ 60+ APIs funcionando



## Arquivos Modificados

- ✓ app/api/checkout/check-payment/route.ts
  - Logs detalhados adicionados
  - Resposta JSON enriquecida com debug
  - Tratamento de erro melhorado

## Próximos Passos

1. 📊 **Analisar os logs** conforme este documento
2. 🛠️ **Identificar o erro específico** (404, 401, etc.)
3. 🛠️ **Aplicar a solução correspondente**
4. ✓ **Validar que o fluxo funciona**

---

Documento criado em: 27/11/2025

Status: ✓ Pronto para Diagnóstico

Build: ✓ Sucesso