

🎯 ANÁLISE COMPLETA: Erro "INVESTMENT not found in enum TransactionType"

📊 SITUAÇÃO ATUAL (Análise Realizada)

✅ Banco Abacus.AI (Local) - PERFEITO

```
-- Enums existentes no banco Abacus.AI:  
TransactionType: INCOME, EXPENSE, INVESTMENT ✅  
CategoryType: INCOME, EXPENSE, INVESTMENT ✅  
  
-- Dados existentes:  
✅ 125+ categorias de investimento criadas  
✅ Todas usando type='INVESTMENT'  
✅ Todas as migrações aplicadas corretamente
```

DATABASE_URL Local:

```
postgresql://  
role_9484b0c23:eaQqYU5eW_gE6aRZJT0XP5sKzkhEA7Q5@db-9484b0c23.db002.hosteddb.reai.io:54  
32/9484b0c23
```

❌ Vercel (Produção) - PROBLEMA IDENTIFICADO

Log do Build (linha 22:41:25):

```
Datasource "db": PostgreSQL database "postgres", schema "public" at "aws-1-sa-  
east-1.pooler.supabase.com:5432"
```

```
Error: P1000: Authentication failed against database server, the provided database cre  
dentials for `postgres` are not valid.
```

Erro de Runtime:

```
PrismaClientUnknownRequestError:  
Invalid prisma.category.findMany() invocation:  
  
Value 'INVESTMENT' not found in enum 'TransactionType'
```

Conclusão:

- ❌ Vercel está tentando conectar ao **Supabase** (credenciais inválidas)
- ❌ Build ignora erro de migração e continua
- ❌ Runtime conecta a um banco **sem INVESTMENT**
- ❌ Todas as rotas que usam categorias falham

ANÁLISE DO CÓDIGO

1. Schema Prisma (✅ Correto)

```
enum TransactionType {
  INCOME      // Receita/Entrada
  EXPENSE     // Despesa/Saída
  INVESTMENT  // Investimento
}

enum CategoryType {
  INCOME      // Categoria de Receita/Entrada
  EXPENSE     // Categoria de Despesa/Saída
  INVESTMENT  // Categoria de Investimento
}
```

2. Migrações (✅ Corretas)

```
20251031191431_add_lgpd_consent
20251031212534_add_investment_category_type
20251031222834_add_investment_to_transaction_type ➡ Adiciona INVESTMENT
```

Conteúdo da migração:

```
ALTER TYPE "TransactionType" ADD VALUE IF NOT EXISTS 'INVESTMENT';
```

3. Script postinstall.sh (⚠ Falha Silenciosa)

```
echo "📦 Aplicando migrações pendentes ao banco de dados..."
npx prisma migrate deploy || echo "⚠ Nenhuma migração pendente ou erro ao aplicar"
```

Problema:

- Tenta aplicar migrações ao Supabase
- Falha porque credenciais são inválidas
- `|| echo` ignora o erro
- Build continua como se nada tivesse acontecido
- Prisma Client é gerado com INVESTMENT
- Runtime conecta a banco SEM INVESTMENT
- **BOOM! 💣 Erro em todas as rotas**

4. Código da Aplicação (✅ Correto)

API Routes:

```
// app/api/categories/route.ts
if (!['INCOME', 'EXPENSE', 'INVESTMENT'].includes(type)) {
  return NextResponse.json({ error: 'Invalid type' }, { status: 400 });
}
```

Webhook:











```
// app/api/webhook/hotmart/route.ts
import { UserStatus } from '@prisma/client';

await prisma.user.update({
  where: { email },
  data: { status: UserStatus.ACTIVE }
});
```

Tudo está usando tipos corretos!

A RAIZ DO PROBLEMA







Comparação de Ambientes:

Componente	Local (Dev)	Vercel (Prod)	Status
DATABASE_URL	Abacus.AI 	Supabase 	DIFERENTE
Schema Prisma	Com INVESTMENT 	Com INVESTMENT 	Igual
Migrações	Aplicadas 	NÃO aplicadas 	DIFERENTE
Enum no Banco	INVESTMENT existe 	INVESTMENT NÃO existe 	DIFERENTE
Dados no Banco	125+ categorias 	Banco vazio/antigo 	DIFERENTE

SOLUÇÃO DEFINITIVA

Opção 1: Usar Banco Abacus.AI em Produção (RECOMENDADO)

Por quê?

-  Banco já está configurado e funcionando
-  Todas as migrações já aplicadas
-  Todos os dados já existem
-  Zero configuração adicional
-  Mesma infraestrutura (Abacus.AI)
-  Simplicidade e consistência

Como:

1. Acessar Vercel:

<https://vercel.com/vinicius-projects-c13a142e/orcamento-planejado/settings/environment-variables>

2. Editar DATABASE_URL:

- Remover: `postgresql://postgres.gvv...@aws-1-sa-east-1.pooler.supabase.com:5432/postgres`

- Adicionar: `postgresql://
role_9484b0c23:eaQqYU5eW_gE6aRZJT0XP5sKzkhEA7Q5@db-9484b0c23.db002.hosteddb.reai.io:
5432/9484b0c23?connect_timeout=15`

3. Aplicar em TODOS os ambientes:

- ☒ Production
- ☒ Preview
- ☒ Development

4. Redeploy:

Deployments → Click no último → ⋮ → Redeploy

Resultado esperado:

☒ Datasource "db": PostgreSQL database "9484b0c23" at "db-9484b0c23.db002.hosteddb.reai.io:5432"
☒ Migrações aplicadas (já existem)
☒ Prisma Client gerado
☒ Build sucesso
☒ Deployment Ready
☒ TUDO FUNCIONA! 🎉

Opção 2: Migrar Supabase (❌ NÃO RECOMENDADO)

Por quê NÃO?

- ❌ Requer configurar novo banco
- ❌ Requer aplicar todas as migrações
- ❌ Requer recriar todos os dados
- ❌ Mais complexo
- ❌ Mais propenso a erros
- ❌ Banco Abacus.AI já funciona perfeitamente

Se ainda assim quiser:

1. Obter credenciais válidas do Supabase
2. Atualizar DATABASE_URL no Vercel com credenciais corretas
3. Garantir que `postinstall.sh` aplique migrações:


```
bash
npx prisma migrate deploy
```
4. Redeploy e aguardar migrações
5. Seed para criar categorias padrão
6. Testar tudo novamente



PLANO DE AÇÃO RECOMENDADO

Passo 1: Decidir Estratégia

Pergunta para você:

- ☐ Usar banco Abacus.AI em produção? (✅ Recomendado - 5 minutos)
 - ☐ Configurar Supabase corretamente? (⚠️ Complexo - 30+ minutos)
-

Se escolher Abacus.AI (Recomendado):

Checklist:

- ☐ Acessar Vercel → Settings → Environment Variables
- ☐ Encontrar DATABASE_URL
- ☐ Editar valor para o banco Abacus.AI
- ☐ Marcar: Production, Preview, Development
- ☐ Salvar
- ☐ Ir em Deployments
- ☐ Redeploy último deployment
- ☐ Aguardar 2-3 minutos
- ☐ Testar: <https://orcamento-planejado.abacusai.app>
- ☒ Celebrar! 🎉

Tempo estimado: 5 minutos

Se escolher Supabase:

Checklist:

- ☐ Criar projeto no Supabase
- ☐ Obter credenciais válidas
- ☐ Atualizar DATABASE_URL no .env local
- ☐ Testar conexão localmente
- ☐ Aplicar todas as migrações localmente
- ☐ Verificar enums no Supabase
- ☐ Criar seed de categorias
- ☐ Atualizar DATABASE_URL no Vercel
- ☐ Redeploy
- ☐ Aguardar migrações aplicarem
- ☐ Testar em produção
- ☒ Resolver erros que surgirem

Tempo estimado: 30-60 minutos

LIÇÕES APRENDIDAS

1. Sempre Validar DATABASE_URL

```
# Local:  
echo $DATABASE_URL  
  
# Vercel:  
Verificar em Settings → Environment Variables
```

2. Não Ignorar Erros de Migração

```
# ❌ Não fazer:  
npx prisma migrate deploy || echo "⚠️ Nenhuma migração pendente ou erro ao aplicar"  
  
# ✅ Fazer:  
npx prisma migrate deploy || (echo "❌ ERRO ao aplicar migrações" && exit 1)
```

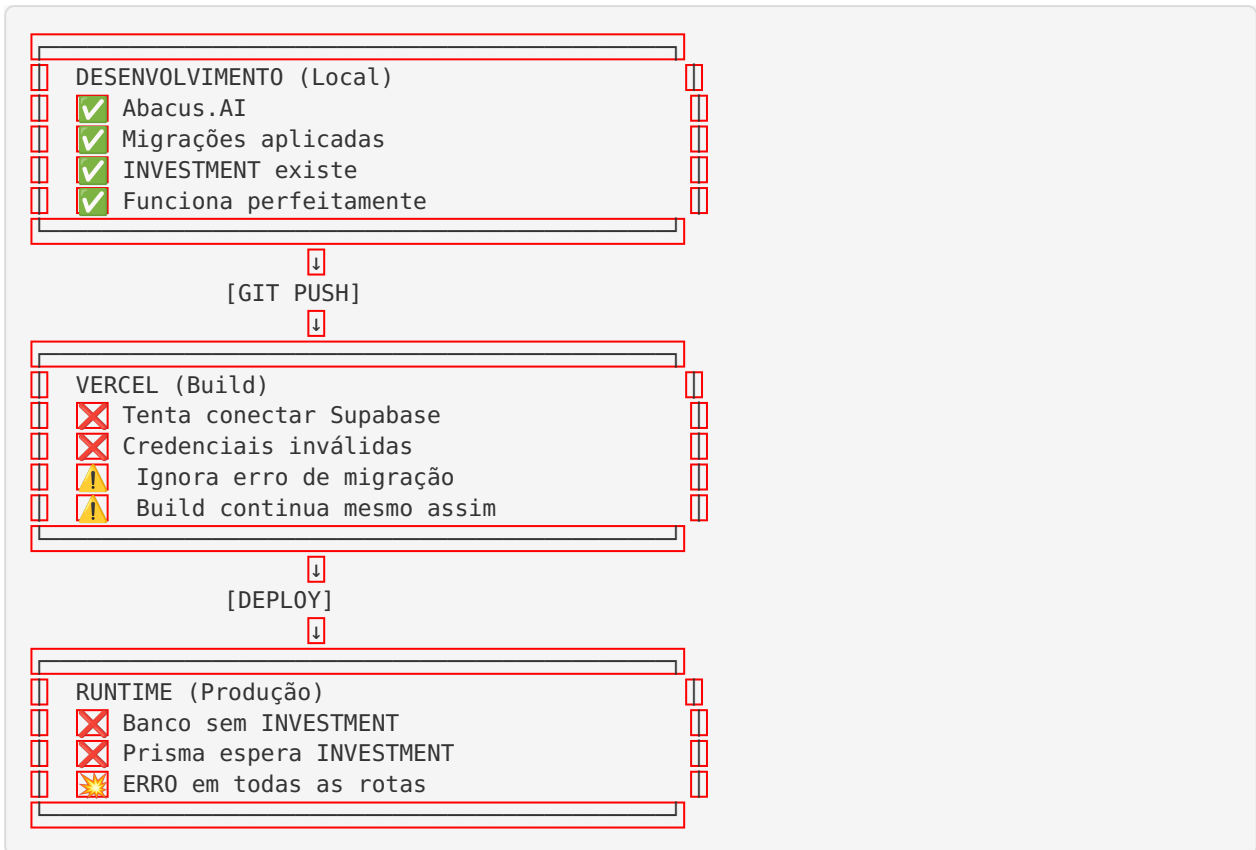
3. Consistência Entre Ambientes

Dev e Prod devem usar:

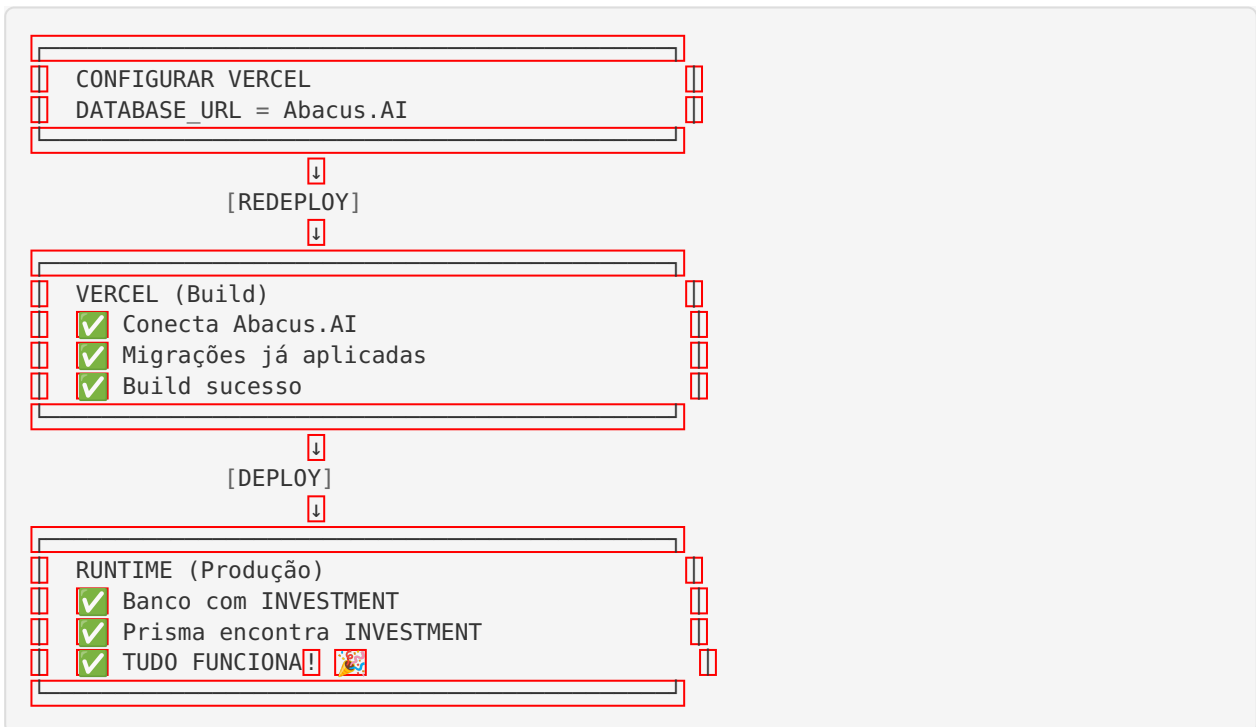
- ✅ Mesmo tipo de banco
 - ✅ Mesma versão
 - ✅ Mesmas migrações aplicadas
 - ✅ Idealmente, mesmo banco (Abacus.AI)
-



DIAGNÓSTICO VISUAL



SOLUÇÃO:



RESUMO EXECUTIVO

O Problema

Vercel está tentando conectar ao Supabase com credenciais inválidas, mas o banco que funciona é o Abacus.AI.


A Solução

Atualizar `DATABASE_URL` no Vercel para apontar ao banco Abacus.AI.

O Resultado

- ✓ Aplicação funcionará perfeitamente em produção
- ✓ Mesmas funcionalidades do ambiente local
- ✓ Zero configuração adicional
- ✓ Zero problemas de migração
- ✓ Zero perda de dados

Tempo

 5 minutos

Dificuldade

 Muito Fácil (copiar e colar)

DECISÃO NECESSÁRIA:

Qual estratégia você prefere?

1.  **Usar banco Abacus.AI em produção** (5 min, fácil, recomendado)
2.  **Configurar Supabase corretamente** (30+ min, complexo)

Me avise sua escolha e vou prosseguir com a implementação!