

✓ SOLUÇÃO DEFINITIVA - DATABASE_URL NO VERCEL

🔍 Causa Raiz Identificada

O arquivo `.env` estava **commitado no repositório GitHub**, mesmo estando no `.gitignore`.

Por Que Isso Causava Erro?

Quando o Vercel fazia o build, a ordem de carregamento das variáveis era:

1. **Primeiro:** Carrega `.env` do repositório GitHub
2. **Depois:** Carrega variáveis do dashboard Vercel

O Prisma **priorizava** as variáveis do arquivo `.env`, que tinha:

```
DATABASE_URL='...?connect_timeout=15' ❌ SEM pgbouncer=true
```

Em vez de usar as variáveis do dashboard que você configurou:

```
DATABASE_URL='...?pgbouncer=true&connect_timeout=15...' ✅ COM pgbouncer=true
```

✓ Solução Aplicada

1 Removido `.env` do Git

```
git rm --cached nextjs_space/.env
```

Resultado:

- ❌ **GitHub:** Não tem mais `.env` no repositório
- ✅ **Local:** Continua tendo `.env` para desenvolvimento
- ✅ **Vercel:** Vai usar APENAS as variáveis do dashboard

2 Commit e Push

```
git commit -m "Remove .env from repository - use Vercel env vars instead"
git push origin main
```

Commit: 5362cb2

3 Vercel Detectará Automaticamente

- O Vercel vai detectar o novo push
- Iniciará um novo deployment em ~30 segundos
- Desta vez usará as variáveis corretas do dashboard



Como Monitorar o Novo Deploy

1. Acesse:

<https://vercel.com/vinicius-projects-c13a142e/orcamento-planejado/deployments>

2. Aguarde Novo Deployment Aparecer

- **Commit:** "Remove .env from repository - use Vercel env vars instead"
- **Status:** Queued → Building → Ready

3. Durante o Build, Monitore os Logs

✓ Sinais de SUCESSO que você DEVE ver agora:

```
Prisma schema loaded from prisma/schema.prisma
Datasource "db": PostgreSQL database "9484b0c23" at "db-9484b0c23.db002.hosteddb.reai.io:5432"
✓ Generated Prisma Client (v6.7.0)
✓ Compiled successfully
```

✗ NÃO deve mais aparecer:

```
aws-1-sa-east-1.pooler.supabase.com ← Supabase (antigo)
Authentication failed ← Erro de autenticação
```

4. Após Deploy Concluir com Sucesso

Teste a aplicação:

1. Clique em "Visit"
2. Faça login: `viniciusleber@gmail.com`
3. Navegue para "Investimentos" → deve carregar normalmente
4. Crie uma transação do tipo "INVESTMENT" → deve funcionar
5. Verifique o Dashboard → gráficos devem carregar



Comparação: Antes vs Depois

Aspecto	Antes (✗ Falhava)	Depois (✓ Funciona)
<code>.env</code> no GitHub	✓ Commitado	✗ Removido
Variáveis Vercel	Ignoradas	Utilizadas
<code>pgbouncer</code> no build	✗ NÃO	✓ SIM
Conexão DB	Falha	Sucesso



Variáveis Corretas no Dashboard Vercel

```

DATABASE_URL = postgresql://
role_9484b0c23:eaQqYU5eW_gE6aRZJT0XP5sKzkhEA7Q5@db-9484b0c23.db002.hosteddb.reai.io:54
32/9484b0c23?pgbouncer=true&connect_timeout=15&pool_timeout=15&connection_limit=10

NEXTAUTH_URL = https://orcamento-planejado.vercel.app
NEXTAUTH_SECRET = 0xm0jVI3lAS8zLqGHeR8MCmqfePU0Aqx
GMAIL_USER = suporteplanejado@gmail.com
GMAIL_APP_PASSWORD = xksbphwgxwqtoyil
ABACUSAI_API_KEY = 6c6cd1cd5406461090e87cb0a37694f9
EMAIL_FROM = Orçamento Planejado <suporteplanejado@gmail.com>

```

Todas marcadas em: Production, Preview, Development



Prevenindo Futuros Erros



Boas Práticas Aplicadas:

- Nunca commitar .env**
 - .env deve estar no .gitignore (já está ✓)
 - Se foi commitado antes, remover com `git rm --cached`
- Variáveis sensíveis apenas no dashboard**
 - Senhas, API keys, tokens → Vercel dashboard
 - Nunca no código ou .env commitado
- Verificação visual antes de deploy**
 - Confirmar variáveis no dashboard
 - Verificar logs de build
 - Testar após deploy
- Ambientes corretos**
 - Marcar Production, Preview E Development
 - Garantir consistência entre ambientes



Resultado Esperado

Após este deployment:

- ✓ App carrega normalmente
- ✓ Investimentos funcionam
- ✓ Sem erros de autenticação
- ✓ Prisma conecta ao Abacus com pooling
- ✓ Performance otimizada para serverless



Checklist Final

Após o deployment concluir:

- [] Status no Vercel: "Ready" (verde)
- [] Logs mostram: db-9484b0c23.db002.hosteddb.reai.io (Abacus)
- [] Logs NÃO mostram: supabase.com

- [] App abre sem erro 500
- [] Login funciona
- [] Investimentos carregam
- [] Transações podem ser criadas
- [] Dashboard mostra gráficos

Se Ainda Houver Erro

Improvável, mas se acontecer:

1. Copie os logs completos do build
2. Verifique se o commit correto foi deployado (5362cb2)
3. Confirme que não há `.env` no repositório (vá no GitHub e verifique)
4. Envie os logs para o DeepAgent

Data: 2025-11-01 03:10 UTC

Status:  Solução aplicada, aguardando deploy

Commit: 5362cb2

Próximo Passo: Monitorar deployment no Vercel