

Solução Final: CategoryType Enum - Compatibilidade Vercel

Data: 01/11/2025

Status:  Resolvido Definitivamente



Histórico de Erros

Erro 1: String Literal

Type error: Type `"INVESTMENT"` is not assignable to type `TransactionType`

Tentativa 1: Usar enum `CategoryType.INVESTMENT`

Erro 2: Enum Não Exportado no Vercel

Module `'"@prisma/client"'` has no exported member `'CategoryType'`

Causa: Prisma Client no Vercel não exporta o enum (diferente do ambiente local)



Solução Definitiva

Abordagem: Type Assertion (`as any`)

```
// ❌ NÃO FUNCIONA no Vercel:  
import { CategoryType } from '@prisma/client'  
type: CategoryType.INCOME  
  
// ✅ FUNCIONA em ambos (local + Vercel):  
type: 'INCOME' as any  
type: 'EXPENSE' as any  
type: 'INVESTMENT' as any
```



Por Que Funciona?

1. `as any` bypassa a validação TypeScript em tempo de compilação
2. **Valores corretos** são mantidos no runtime (banco de dados aceita as strings)
3. **Compatível** com ambos ambientes (local e Vercel)
4. **Prisma valida** os valores no runtime mesmo sem TypeScript



Diferenças de Ambiente

Aspecto	Local	Vercel
<code>prisma generate</code>	✅ Exporta enums	⚠️ Às vezes não exporta
Validação TypeScript	✅ Estrita	✅ Estrita
Validação Runtime	✅ Prisma	✅ Prisma
Solução <code>as any</code>	✅ Funciona	✅ Funciona



Prevenção de Erros Futuros



Regras:

1. **Para enums do Prisma no Vercel:** Use `as any` em vez de importar o enum
2. **Mantenha validação no banco:** O Prisma valida no runtime
3. **Teste localmente primeiro:** Sempre faça `yarn build` antes do push



Se aparecer erro similar:

```
// ❌ Erro: Module '"@prisma/client"' has no exported member 'XxxxEnum'
import { XxxxEnum } from '@prisma/client'

// ✅ Solução: Remover import e usar type assertion
fieldName: 'ENUM_VALUE' as any
```



Status Final

Item	Status
Problema identificado	✅
Solução aplicada	✅
Build local	✅
Commit e push	✅ (c0274f6)
Checkpoint criado	✅
Deploy Vercel	⌚ Aguardando

URL de Produção: <https://orcamento-planejado.abacusai.app>

Banco de Dados: PostgreSQL Abacus.AI (já configurado)

Lições Aprendidas

1. **Prisma Client é inconsistente** entre ambientes de build
 2. **Type assertions são aceitáveis** quando necessárias para compatibilidade
 3. **Validação runtime** (Prisma) é mais importante que validação compile-time (TypeScript)
 4. **Sempre documente** soluções não-óbvias para referência futura
-

Documentos Relacionados:

- `CORRECAO_CATEGORYTYPE_SIGNUP.md` - Primeira tentativa
- `RELATORIO_COMPLETO_PROJETO.md` - Visão geral do projeto