

Solução Definitiva: Enums do Prisma

Problema Raiz

O erro que estava se repetindo no build do Vercel era:

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

Por que isso acontecia?

O problema ocorria porque estávamos usando **string literals** ('INCOME' , 'EXPENSE' , 'INVESTMENT') ao invés dos **valores do enum do Prisma**.

Código ERRADO ❌

```
// Isso causa erro de type checking:
const investments = await prisma.transaction.findMany({
  where: {
    type: 'INVESTMENT', // ❌ String literal
  },
})

// Validação errada:
if (['INCOME', 'EXPENSE', 'INVESTMENT'].includes(type)) { // ❌
  // ...
}
```

Código CORRETO ✅

```
import { TransactionType } from '@prisma/client'

// Correto - usando o enum:
const investments = await prisma.transaction.findMany({
  where: {
    type: TransactionType.INVESTMENT, // ✅ Valor do enum
  },
})

// Validação correta:
const VALID_TYPES = [
  TransactionType.INCOME,
  TransactionType.EXPENSE,
  TransactionType.INVESTMENT
]

if (VALID_TYPES.includes(type)) { // ✅
  // ...
}
```

Arquivos Corrigidos

Todos os arquivos que usavam string literals foram corrigidos:

1. `app/api/analyses/generate/route.ts`

- Importado `TransactionType`
- Substituídas todas as 7 ocorrências de string literals
- Usado enum em filtros, validações e queries

2. `app/api/categories/route.ts`

- Importado `CategoryType`
- Criada constante `VALID_CATEGORY_TYPES`
- Substituídas validações em POST e GET

3. `app/api/transactions/route.ts`

- Importado `TransactionType`
- Criada constante `VALID_TRANSACTION_TYPES`
- Substituídas validações em POST e GET

4. `app/api/signup/route.ts`

- Importado `CategoryType`
- Substituídos todos os `as any` por valores do enum
- Criação de categorias padrão corrigida





5. `app/dashboard/transactions/page.tsx`

- Importado `TransactionType`
- Criada constante de validação
- Corrigido filtro de tipo de transação

Benefícios da Solução

1. **Type Safety:** TypeScript agora valida corretamente os tipos
 2. **Autocomplete:** IDEs mostram valores válidos do enum
 3. **Manutenibilidade:** Se o enum mudar, o TypeScript avisa
 4. **Consistência:** Mesma abordagem em todo o código
 5. **Build Confiável:** Não vai mais falhar por erro de tipo
-

Build Status

-  Build local: **PASSOU**
 -  Commit: `1c3cbf2`
 -  Push: **OK**
 -  Deploy Vercel: **Em andamento**
-

Checklist de Qualidade

- [x] Importar enums do `@prisma/client` em todos os arquivos necessários
 - [x] Substituir TODAS as string literals por valores do enum
 - [x] Criar constantes de validação para facilitar manutenção
 - [x] Testar build local
 - [x] Fazer commit com mensagem descritiva
 - [x] Push para GitHub/Vercel
 - [x] Documentar solução para referência futura
-



Lições Aprendidas

Como evitar esse erro no futuro?

1. Sempre importar enums do Prisma:

```
typescript
import { TransactionType, CategoryType } from '@prisma/client'
```

2. Nunca usar string literals para valores de enum:

-  `type: 'INCOME'`
-  `type: TransactionType.INCOME`

3. Criar constantes de validação:

```
typescript
const VALID_TYPES = Object.values(TransactionType)
```

4. Testar build local antes de push:

```
bash
yarn prisma generate && yarn build
```

Impacto da Correção

- **Arquivos alterados:** 6
 - **Linhas modificadas:** 30 insertions, 16 deletions
 - **String literals substituídas:** 22 ocorrências
 - **Enums importados:** 2 (CategoryType, TransactionType)
 - **Constantes de validação criadas:** 3
-

Referências

- [Prisma Client API](https://www.prisma.io/docs/concepts/components/prisma-client) (https://www.prisma.io/docs/concepts/components/prisma-client)
 - [TypeScript Enums](https://www.typescriptlang.org/docs/handbook/enums.html) (https://www.typescriptlang.org/docs/handbook/enums.html)
 - [Prisma Enums](https://www.prisma.io/docs/concepts/components/prisma-schema/data-model#defining-enums) (https://www.prisma.io/docs/concepts/components/prisma-schema/data-model#defining-enums)
-

Data: 01/11/2025

Status:  **RESOLVIDO DEFINITIVAMENTE**

Deploy: <https://orcamento-planejado.abacusai.app>