

Implementação: Categorias Padrão + Sofia Melhorada

Data: 01/11/2025

Status:  Implementado com Sucesso




Resumo das Implementações

1. Categorias Padrão Melhoradas

Problema Original:

- Categorias padrão não estavam sendo criadas automaticamente para novos usuários
- Screenshot mostrava “Nenhuma categoria de despesa”

Solução Implementada:

-  Criação em lote usando `Promise.all` (mais eficiente)
-  Melhor tratamento de erros com logs específicos
-  Try-catch isolado para não quebrar o signup se categorias falharem

Categorias Criadas:

Receitas (4):

- Salário (#00bf63)
- Freelance (#20c997)
- Bonificação (#6f42c1)
- Renda Extra (#28a745)

Despesas (5):

- Moradia (#6c757d)
- Transporte (#ffc107)
- Mercado (#fd7e14)
- Alimentação (#dc3545)
- Saúde (#e83e8c)

Investimentos (5):

- Renda Fixa (#00bf63)
- Fundos (#6f42c1)
- Ações (#20c997)
- Cripto (#ffc107)
- Outros (#737373)

Total: 14 categorias padrão criadas automaticamente

3. Sofia (IA) Melhorada

Mudanças Implementadas:

1. Uso de “R\$” e “reais”

- ❌ Antes: `total_mes`, `media_semanal_estimada` (sem unidade clara)
- ✅ Agora: `total_mes_reais`, `media_semanal_reais` (com unidade explícita)
- Sofia agora sempre fala em “R\$” ou “reais” nas análises

2. Análise de Metas

- ✅ Busca metas do usuário no banco
- ✅ Calcula progresso percentual de cada meta
- ✅ Avalia se o usuário está no caminho certo
- ✅ Sugere ajustes realistas

3. Análise de Diversificação de Investimentos

- ✅ Busca todos os investimentos do usuário
- ✅ Agrupa por categoria (Renda Fixa, Ações, Fundos, Cripto, Outros)
- ✅ Calcula percentual de cada categoria
- ✅ Avalia diversificação da carteira
- ✅ Sugere melhorias se necessário

4. Texto Mais Breve

- ❌ Antes: ~220-280 palavras
- ✅ Agora: ~180-220 palavras
- Prompt mais direto e objetivo
- `max_tokens` reduzido de 800 → 600

Arquivos Modificados

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

Mudanças:

```
// ✅ Criação em lote (Promise.all)
const allCategories = [...incomeCategories, ...expenseCategories, ...investmentCategories]

await Promise.all(
  allCategories.map(categoryData =>
    prisma.category.create({ data: categoryData })
  )
)

// ✅ Log de sucesso
console.log(`✅ ${allCategories.length} categorias padrão criadas para usuário ${user.id}`)

// ✅ Try-catch isolado
try {
  // criação de categorias
} catch (categoryError) {
  console.error('⚠️ Erro ao criar categorias padrão:', categoryError)
  // Não falhar o signup se categorias falharem
}
```

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

Mudanças:

```
// ✅ Buscar metas
const goals = await prisma.goal.findMany({
  where: { userId },
  select: {
    name: true,
    targetAmount: true,
    progress: true,
  },
})

// ✅ Buscar investimentos
const investments = await prisma.transaction.findMany({
  where: {
    userId,
    type: 'INVESTMENT',
  },
  include: {
    category: true,
  },
})

// ✅ Calcular diversificação
const investmentDiversification = Object.entries(investmentsByCategory).map(([categoria, total]) => ({
  categoria,
  total,
  percentual: totalInvestments > 0 ? ((total / totalInvestments) * 100).toFixed(1) : 0,
}))

// ✅ Payload com metas e investimentos
const payload = {
  referencia: { ano, mes },
  renda: {
    total_mes_reais: income, // ✅ "_reais"
    media_semanal_reais: weeklyIncomeAvg,
    media_diaria_reais: dailyIncomeAvg,
  },
  gastos: { ... },
  saldo: { ... },
  metas: goals.length > 0 ? goals.map(g => ({ // ✅ Novo!
    nome: g.name,
    meta_reais: Number(g.targetAmount),
    atual_reais: Number(g.progress),
    progresso_percentual: ...
  })) : undefined,
  investimentos: totalInvestments > 0 ? { // ✅ Novo!
    total_reais: totalInvestments,
    diversificacao: investmentDiversification,
  } : undefined,
}

// ✅ Prompt mais breve e direto
const systemPrompt = `Você é Sofia, consultora financeira empática e prática. Estilo: direto, claro e acolhedor.`
```

Diretrizes:

1. **Visão geral**: Compare renda e gastos em R\$ (use "R\$" ou "reais"), apresente o saldo mensal.
2. **Padrões**: Identifique categorias de maior gasto e outliers (se houver).
3. **Metas**: Se houver metas definidas, avalie o progresso e sugira ajustes reais-

tas.

4. ****Investimentos****: Se houver investimentos, comente a diversificação da carteira (Renda Fixa, Ações, Fundos, Cripto) e sugira melhorias se necessário.
5. ****Ações****: Proponha 2-3 micro-ajustes práticos (ex: reduzir 10% em uma categoria, criar reserva semanal).
6. ****Encerramento****: Recapitule em 3-4 bullets curtos. Feche com uma frase motivacional.

Limites:

- Use sempre "R\$" ou "reais" para valores monetários.
- Evite jargões técnicos e recomendações de produtos específicos.
- Resposta breve: ~180–220 palavras máximo.`

Testes Realizados

Build Local:

```
cd /home/ubuntu/orcamento_planejado/nextjs_space
yarn build
```

Resultado:  Sucesso (exit_code=0)

Deploy:

```
git add -A
git commit -m "feat: Categorias padrão melhoradas + Sofia com análise de metas e investimentos"
git push origin main
```

Resultado:  Commit 5a4ab22 | Push concluído





Checkpoint:

```
build_and_save_nextjs_project_checkpoint
```

Resultado:  “Categorias padrão + Sofia melhorada”




Benefícios

Categorias Padrão:

-  Melhor onboarding (usuários não precisam criar categorias do zero)
-  Organização automática desde o início
-  Criação mais rápida (Promise.all)
-  Logs claros para debugging

Sofia Melhorada:

-  Linguagem mais clara e acessível (“R\$” ao invés de “unidades”)

-  Análise mais completa (metas + investimentos)
-  Respostas mais breves e diretas
-  Recomendações mais personalizadas

Comportamento Esperado

Para Novos Usuários:

1. Ao se cadastrar, 14 categorias padrão são criadas automaticamente
2. Podem editar ou excluir qualquer categoria
3. Se excluir categoria usada, transações vão para “Desconhecida”

Para Análises (Sofia):

1. Sofia menciona valores sempre em “R\$” ou “reais”
2. Se houver metas, Sofia avalia progresso e sugere ajustes
3. Se houver investimentos, Sofia analisa diversificação
4. Respostas mais curtas e práticas (~180-220 palavras)

Status Final

Item	Status
Categorias padrão implementadas	
Sofia com “R\$”	
Sofia analisa metas	
Sofia analisa investimentos	
Sofia mais breve	
Build local	
Commit e push	 (5a4ab22)
Checkpoint criado	
Deploy Vercel	 Aguardando

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

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

Observações Importantes

Usuários Antigos:

- ⚠️ Usuários já cadastrados **não** receberão as categorias retroativamente
- Solução: Eles podem criar manualmente ou você pode criar um script de migração

Logs:

- ✅ Console agora mostra: ✅ 14 categorias padrão criadas para usuário {userId}
- ⚠️ Se falhar: ⚠️ Erro ao criar categorias padrão: {erro}
- Signup **não falha** se categorias falharem (usuário pode criar depois)

Sofia:

- ✅ Só analisa metas se o usuário tiver metas criadas
- ✅ Só analisa investimentos se houver transações tipo INVESTMENT
- ✅ Se não houver dados, Sofia trabalha apenas com renda/gastos/saldo

Documentos Relacionados:

- SOLUCAO_FINAL_CATEGORYTYPE.md - Correção anterior (enum)
- RELATORIO_COMPLETO_PROJETO.md - Visão geral do projeto