



Otimizações do Core Financeiro

Implementadas em: 31 de outubro de 2025



Resumo

Este documento descreve as três principais otimizações implementadas no sistema financeiro do **Orçamento Planejado**, focadas em melhorar a experiência do usuário na gestão de categorias, transações e exportação de dados.



1. Categorias Padrão e Comportamento Dinâmico



Implementação

Categorias Padrão Criadas Automaticamente

Quando um novo usuário se registra, o sistema cria automaticamente **10 categorias padrão**:



Receitas (5 categorias):

1. **Salário** - Verde (#00bf63)
2. **Bonificações** - Verde claro (#20c997)
3. **Freelance** - Azul (#17a2b8)
4. **Investimentos** - Roxo (#6f42c1)
5. **Outros** - Verde escuro (#28a745)



Despesas (5 categorias):

1. **Moradia** - Cinza (#6c757d)
2. **Alimentação** - Vermelho (#dc3545)
3. **Transporte** - Amarelo (#ffc107)
4. **Saúde** - Rosa (#e83e8c)
5. **Lazer** - Laranja (#fd7e14)

Edição de Categorias



Nome editável: Ao editar o nome de uma categoria, todas as transações associadas são automaticamente atualizadas.



Validação: Não permite categorias duplicadas por tipo (receita/despesa).



Cor personalizável: Cada categoria pode ter uma cor diferente para facilitar visualização.

Exclusão Inteligente



Migração automática: Quando uma categoria é excluída, todas as transações associadas são automaticamente migradas para uma categoria chamada **"Desconhecida"** do mesmo tipo.



Sem perda de dados: Nenhuma transação é perdida durante o processo de exclusão.



Criação automática: A categoria "Desconhecida" é criada automaticamente se não existir.

Arquivos Modificados

- `scripts/seed.ts` - Adicionadas 5 categorias de cada tipo
- `app/api/categories/[id]/route.ts` - Lógica de migração para “Desconhecida”

2. Caixa de Seleção Inteligente de Categorias

Implementação

Criamos um componente **CategoryCombobox** que substitui o select tradicional por uma interface moderna e inteligente.

Funcionalidades

Busca Incremental

- Digite para filtrar categorias existentes
- Busca em tempo real enquanto você digita
- Destaque visual da categoria selecionada

Scroll Vertical

- Altura máxima de 300px
- Scroll automático quando há muitas categorias
- Interface limpa e organizada

Criação On-the-Fly

- Se a categoria não existe, aparece botão `+ Criar "[nome digitado]"`
- Cria a categoria instantaneamente
- Seleciona automaticamente a nova categoria
- Toast de confirmação de sucesso

Visual Moderno

- Cores das categorias visíveis na lista
- Ícones intuitivos
- Design consistente com o tema do app
- Animações suaves

Exemplo de Uso

```
<CategoryCombobox
  categories={categories}
  value={formData.categoryId}
  onChange={(value) => setFormData(prev => ({ ...prev, categoryId: value })))}
  type={formData.type}
  onCategoryCreated={() => router.refresh()}
/>
```

Arquivos Criados/Modificados

- `components/categories/category-combobox.tsx` - **NOVO** componente inteligente
- `components/transactions/transaction-form.tsx` - Integração do combobox

3. Exportação de Extratos Mensais

Implementação

Adicionado sistema completo de exportação de transações em dois formatos: **Excel** e **PDF**.

Funcionalidades

Exportação em Excel (.xlsx)



- Planilha formatada com colunas: Data | Descrição | Tipo | Categoria | Valor
- Valores formatados em reais (R\$)
- Datas em formato brasileiro (DD/MM/AAAA)
- Nome do arquivo: `extrato-[mês]-[ano].xlsx`

Exportação em PDF

- Tabela profissional com cabeçalho
- Título com período selecionado
- Totais calculados automaticamente:
- Total de Receitas
- Total de Despesas
- Saldo (Receitas - Despesas)
- Design limpo e profissional
- Nome do arquivo: `extrato-[mês]-[ano].pdf`

Interface do Usuário

Botão “Exportar” na página de transações

- Dropdown com opções Excel e PDF
- Ícones visuais ( Excel,  PDF)
- Feedback de “Exportando...”
- Toast de sucesso/erro
- Fecha automaticamente ao clicar fora

Filtros de Período

 **Exportação por mês:** Atualmente exporta o mês corrente

 **Extensível:** A API suporta filtros personalizados de `month` e `year`

Arquivos Criados/Modificados

- `app/api/transactions/export/route.ts` - **NOVA** API de exportação
 - `components/transactions/transactions-client.tsx` - Botão e lógica de exportação
 - `package.json` - Adicionadas bibliotecas `xlsx`, `jspdf`, `jspdf-autotable`
-



Bibliotecas Adicionadas

```
{
  "dependencies": {
    "xlsx": "^0.18.5",      // Exportação Excel
    "jspdf": "^3.0.3",      // Geração de PDF
    "jspdf-autotable": "^5.0.2" // Tabelas no PDF
  }
}
```



Estrutura da API de Exportação

Endpoint

GET /api/transactions/**export**

Query Parameters

Parâmetro	Tipo	Descrição	Exemplo
format	string	Formato do arquivo (xlsx ou pdf)	xlsx
month	number	Mês (1-12)	10
year	number	Ano (YYYY)	2025

Exemplo de Requisição

```
const response = await fetch(
  '/api/transactions/export?format=xlsx&month=10&year=2025'
)
```

Response

- **Status 200:** Arquivo binário (Excel ou PDF)
- **Status 404:** Nenhuma transação encontrada
- **Status 401:** Não autorizado
- **Status 500:** Erro interno



Melhorias de UX

Transações

- ✓ **Formulário mais intuitivo:** Combobox substituiu o select tradicional
- ✓ **Criação rápida:** Não precisa mais sair da página para criar categorias

- ✓ **Feedback visual:** Cores e ícones melhoram a identificação

Categorias

- ✓ **Exclusão segura:** Dados nunca são perdidos
- ✓ **Edição em cascata:** Mudanças refletem automaticamente
- ✓ **Cores personalizadas:** Facilitam a visualização nos gráficos

Exportação

- ✓ **Dois formatos:** Excel para análise, PDF para compartilhamento
- ✓ **Download rápido:** Geração instantânea de arquivos
- ✓ **Nomes descritivos:** Arquivos com data no nome

Testando as Funcionalidades

1. Teste de Categorias Padrão

1. Crie um novo usuário
2. Faça login
3. Navegue até “Categorias”
4. Verifique que existem 10 categorias (5 receitas + 5 despesas)

2. Teste de Combobox Inteligente

1. Vá em “Nova Transação”
2. Digite um texto no campo de categoria
3. Verifique a busca incremental
4. Digite um nome inexistente
5. Clique em “+ Criar [nome]”
6. Verifique que a categoria foi criada e selecionada

3. Teste de Exclusão de Categoria

1. Crie algumas transações com uma categoria específica
2. Exclua essa categoria
3. Verifique que as transações foram migradas para “Desconhecida”
4. Confirme que nenhum dado foi perdido

4. Teste de Exportação

1. Vá em “Transações”
 2. Clique em “Exportar”
 3. Escolha “Excel” ou “PDF”
 4. Verifique o download do arquivo
 5. Abra o arquivo e confirme os dados
-

Deploy

As otimizações foram:

- ☒ Commitadas no Git
- ☒ Enviadas para o GitHub
- ☒ Checkpoint criado
- ☒ Build bem-sucedido

Próximos Passos no Vercel

O Vercel irá automaticamente:

1. Detectar o novo commit
2. Fazer o build da aplicação
3. Fazer deploy da nova versão
4. Disponibilizar em produção

Notas Técnicas

Prisma Schema

A estrutura do modelo `Category` já suportava todas as funcionalidades:

```
model Category {
  id      String @id @default(cuid())
  name    String
  type    TransactionType // INCOME ou EXPENSE
  color   String?
  userId  String

  user    User      @relation(...)
  transactions Transaction[]

  @@unique([userId, name, type])
}
```

Validações Implementadas

- ☒ Categoria duplicada (mesmo nome + tipo + usuário)
- ☒ Nome vazio ou apenas espaços
- ☒ Tipo inválido (apenas INCOME ou EXPENSE)
- ☒ Autorização (apenas o dono pode modificar)

Compatibilidade

Usuários Existentes

- ☒ **Totalmente compatível:** Usuários existentes não são afetados
- ☒ **Sem migração necessária:** Categorias antigas continuam funcionando
- ☒ **Adoção gradual:** Podem criar novas categorias com o combobox

Dados Históricos

- ✓ **Preservados:** Todas as transações antigas permanecem intactas
 - ✓ **Editáveis:** Podem ser atualizadas para usar o novo combobox
-



Referências

Componentes Utilizados

- **Radix UI:** Popover, Command (combobox)
- **Lucide React:** Ícones
- **react-hot-toast:** Notificações
- **Next.js:** Routing e API

Bibliotecas de Exportação

- **XLSX:** [SheetJS](https://docs.sheetjs.com/) (https://docs.sheetjs.com/)
 - **jsPDF:** [jsPDF Documentation](https://artskydj.github.io/jsPDF/docs/) (https://artskydj.github.io/jsPDF/docs/)
 - **jsPDF-AutoTable:** [AutoTable Plugin](https://github.com/simonbengtsson/jsPDF-AutoTable) (https://github.com/simonbengtsson/jsPDF-AutoTable)
-



Checklist de Implementação

- [x] Atualizar seed.ts com 5 categorias de cada tipo
 - [x] Implementar lógica de migração para “Desconhecida”
 - [x] Criar componente CategoryCombobox
 - [x] Integrar combobox no formulário de transações
 - [x] Instalar bibliotecas de exportação
 - [x] Criar API de exportação (/api/transactions/export)
 - [x] Adicionar botão de exportação na UI
 - [x] Implementar download de Excel
 - [x] Implementar download de PDF
 - [x] Adicionar feedback visual (toasts)
 - [x] Testar build de produção
 - [x] Commit e push para GitHub
 - [x] Criar checkpoint
-



Conclusão

As três otimizações implementadas melhoram significativamente a experiência do usuário:

1. **Categorias Padrão:** Usuários novos já começam com estrutura organizada
2. **Combobox Inteligente:** Criação e seleção de categorias mais rápida e intuitiva
3. **Exportação:** Facilita análise externa e compartilhamento de dados

Todas as funcionalidades foram testadas e estão prontas para produção! 🚀