




# RELATÓRIO COMPLETO DO PROJETO - Orçamento Planejado

---

**Data:** 2025-11-01

**Status:**  Produção - Totalmente Funcional

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

**Repositório GitHub:** <https://github.com/lebervinicius-dev/orcamento-planejado>

---



## VISÃO GERAL

---

**Nome:** Orçamento Planejado

**Tipo:** SaaS de Gestão Financeira Pessoal

**Plataformas:** Web (PWA) + Android (futuramente)

**Stack:** Next.js 14, TypeScript, Prisma, PostgreSQL, NextAuth.js

**Monetização:** Hotmart (R\$ 14,90/mês)

---



## ARQUITETURA TÉCNICA

---

### Frontend

- **Framework:** Next.js 14.2.28 (App Router)
- **Linguagem:** TypeScript 5.2.2
- **UI:** React 18.2.0 + Tailwind CSS 3.3.3
- **Componentes:** Shadcn UI (Radix UI)
- **Gráficos:** Chart.js 4.4.9 + React-Chartjs-2
- **Validação:** React Hook Form + Zod
- **Estado:** Zustand (se necessário)

### Backend

- **API:** Next.js API Routes
- **Autenticação:** NextAuth.js 4.24.11 (Credentials Provider)
- **ORM:** Prisma 6.7.0
- **Banco de Dados:** PostgreSQL (Abacus.AI Hosted)
- **Email:** Nodemailer (Gmail SMTP)

### Infraestrutura

- **Hosting:** Vercel (Frontend + API)
  - **Banco de Dados:** Abacus.AI PostgreSQL (db-9484b0c23)
  - **Domínio:** [orcamento-planejado.abacusai.app](https://orcamento-planejado.abacusai.app)
  - **Repositório:** GitHub (lebervinicius-dev/orcamento-planejado)
-

## AUTENTICAÇÃO E USUÁRIOS

---

### Sistema de Autenticação

- **Provider:** NextAuth.js com Credentials
- **Método:** Email + Senha (bcrypt)
- **Sessão:** JWT (stateless)
- **Proteção:** Middleware para rotas protegidas

### Funcionalidades

- ✓ Registro de usuário
- ✓ Login com email/senha
- ✓ Logout
- ✓ Recuperação de senha por email
- ✓ Troca de senha no primeiro login
- ✓ Modal de consentimento LGPD
- ✓ Termos de uso

### Roles

- **user:** Usuário padrão (acesso ao dashboard)
- **admin:** Administrador (acesso ao painel admin)

### Status de Usuário (UserStatus enum)

- **PENDING:** Aguardando confirmação
- **ACTIVE:** Ativo e com acesso
- **TRIAL:** Período de teste
- **SUSPENDED:** Suspenso temporariamente
- **CANCELED:** Cancelado/bloqueado

### Usuários de Teste

```
// Usuário Padrão
Email: teste@teste.com
Senha: teste123
Role: user

// Administrador
Email: admin@teste.com
Senha: admin123
Role: admin
```

---

## BANCO DE DADOS

---

### Informações de Conexão

**Provider:** Abacus.AI PostgreSQL  
**Host:** db-9484b0c23.db002.hosteddb.reai.io  
**Port:** 5432  
**Database:** 9484b0c23  
**User:** role\_9484b0c23

**DATABASE\_URL (Produção - Com Pooling):**

```
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
```

⚠ **IMPORTANTE:** A URL inclui `pgbouncer=true` para compatibilidade com serverless (Vercel).

**Schema Principal (Modelos Prisma)****1. User (Usuário)**

```
- id: String (cuid)
- name: String
- email: String (unique)
- password: String (bcrypt)
- phone: String?
- role: String (user/admin)
- status: UserStatus (ACTIVE, PENDING, TRIAL, SUSPENDED, CANCELED)
- isActive: Boolean (legado - usar status)
- hotmartId: String? (unique)
- resetToken: String?
- resetTokenExpiry: DateTime?
- firstLogin: Boolean (default: true)
- lgpdConsentAt: DateTime?
- canceledAt: DateTime?
- createdAt: DateTime
- updatedAt: DateTime
```

**2. Category (Categoria)**

```
- id: String (cuid)
- name: String
- type: CategoryType (INCOME, EXPENSE, INVESTMENT)
- color: String? (hex)
- userId: String
- createdAt: DateTime
- updatedAt: DateTime
```

Unique: [userId, name, type]

**3. Transaction (Transação)**

```
- id: String (cuid)
- amount: Decimal (12,2)
- description: String
- date: DateTime
- type: TransactionType (INCOME, EXPENSE, INVESTMENT)
- categoryId: String
- userId: String
- isRecurring: Boolean (default: false)
- recurrenceType: RecurrenceType? (MONTHLY, WEEKLY, YEARLY, CUSTOM)
- createdAt: DateTime
- updatedAt: DateTime
```

## 4. Investment (Investimento)

```
- id: String (cuid)
- name: String
- type: InvestmentType (STOCKS, CRYPTO, REAL_ESTATE, etc)
- initialAmount: Decimal (12,2)
- currentAmount: Decimal (12,2)
- profitPercentage: Decimal (5,2)
- startDate: DateTime
- userId: String
- notes: String?
- createdAt: DateTime
- updatedAt: DateTime
```

## 5. Goal (Meta)

```
- id: String (cuid)
- name: String
- targetAmount: Decimal (12,2)
- currentAmount: Decimal (12,2) (default: 0)
- deadline: DateTime
- userId: String
- completed: Boolean (default: false)
- createdAt: DateTime
- updatedAt: DateTime
```

## 6. AiAnalysis (Análise IA)

```
- id: String (cuid)
- userId: String
- month: Int (1-12)
- year: Int
- analysis: String (JSON/Text)
- createdAt: DateTime
```

## Enums Importantes

```
enum UserStatus {
    PENDING
    ACTIVE
    TRIAL
    SUSPENDED
    CANCELED
}

enum CategoryType {
    INCOME      // Receita
    EXPENSE      // Despesa
    INVESTMENT   // Investimento
}

enum TransactionType {
    INCOME      // Receita/Entrada
    EXPENSE      // Despesa/Saída
    INVESTMENT   // Investimento
}

enum RecurrenceType {
    MONTHLY
    WEEKLY
    YEARLY
    CUSTOM
}

enum InvestmentType {
    STOCKS      // Ações
    CRYPTO       // Criptomoedas
    REAL_ESTATE  // Imóveis
    FIXED_INCOME // Renda Fixa
    FUNDS        // Fundos
    OTHER        // Outros
}
```

## Migrações Aplicadas

1. ☒ 20251031191431\_add\_lgpd\_consent - Campo de consentimento LGPD
2. ☒ 20251031212534\_add\_investment\_category\_type - Tipo INVESTMENT em CategoryType
3. ☒ 20251031222834\_add\_investment\_to\_transaction\_type - Tipo INVESTMENT em TransactionType



## VARIÁVEIS DE AMBIENTE (.env)

### Banco de Dados

```
DATABASE_URL='postgresql://
role_9484b0c23:eaQqYU5eW_gE6aRZJT0XP5sKzkhEA7Q5@db-9484b0c23.db002.hosteddb.reai.io:
5432/9484b0c23?pgbouncer=true&connect_timeout=15&pool_timeout=15&connection_limit=10'
```

## NextAuth

```
NEXTAUTH_SECRET=0xm0jVI3lAS8zLqGHeR8MCmqfePU0Aqx
NEXTAUTH_URL=https://orcamento-planejado.abacusai.app
```

## Gmail (Recuperação de Senha)

```
GMAIL_USER=suporteplanejado@gmail.com
GMAIL_APP_PASSWORD=xksbphwgxwqtoyil
```

## Hotmart (Webhook e Checkout)

```
HOTMART_CHECKOUT_URL=https://pay.hotmart.com/V102667493J
HOTMART_WEBHOOK_SECRET=q9PZsmnMQ2gkfcrcosnMCLGxZvcWdq26170fd1-1f94-4e32-b7de-efdd-f5f7824
NEXT_PUBLIC_HOTMART_CHECKOUT_URL=https://pay.hotmart.com/V102667493J
```

## LLM API (Análises com IA)

```
ABACUSAI_API_KEY=6c6cd1cd5406461090e87cb0a37694f9
```



## DEPLOY E BUILD

### Configuração Vercel (vercel.json)

```
{
  "buildCommand": "node scripts/check-db-url.js && prisma migrate deploy && prisma generate && npm run build"
}
```

#### Ordem correta dos comandos:

1. `check-db-url.js` - Valida DATABASE\_URL
2. `prisma migrate deploy` - Aplica migrações
3. `prisma generate` - Gera Prisma Client
4. `npm run build` - Build do Next.js

**CRÍTICO:** Esta ordem é essencial! Gerar o Prisma Client ANTES de aplicar migrações causa erros de enum.

### Scripts Disponíveis

```
{
  "dev": "next dev",
  "build": "next build",
  "start": "next start",
  "lint": "next lint",
  "postinstall": "prisma generate"
}
```

## GitHub Workflow

**Branch:** main

**Auto-deploy:** ☒ Ativado no Vercel

**Quando:** A cada push na branch main

---



## FUNCIONALIDADES IMPLEMENTADAS

---

### 1. Autenticação e Usuário

- [x] Cadastro de usuário
- [x] Login com email/senha
- [x] Logout
- [x] Recuperação de senha por email
- [x] Troca de senha obrigatória no primeiro login
- [x] Modal de consentimento LGPD
- [x] Modal de termos de uso
- [x] Perfil do usuário (drawer lateral)
- [x] Edição de dados do usuário

### 2. Dashboard

- [x] Resumo financeiro (receitas, despesas, saldo)
- [x] Gráficos de pizza (categorias)
- [x] Gráficos de barras (evolução mensal)
- [x] Filtros por período
- [x] Exportação de dados

### 3. Transações

- [x] Listagem de transações
- [x] Criação de transação (receita/despesa/investimento)
- [x] Edição de transação
- [x] Exclusão de transação
- [x] Filtros (tipo, categoria, período)
- [x] Busca por descrição
- [x] Transações recorrentes
- [x] Exportação CSV

### 4. Categorias

- [x] Listagem de categorias por tipo
- [x] Criação de categoria customizada
- [x] Edição de categoria
- [x] Exclusão de categoria
- [x] Cores personalizadas
- [x] Categorias padrão (seed)

### 5. Investimentos

- [x] Listagem de investimentos

- [x] Criação de investimento
- [x] Edição de investimento
- [x] Exclusão de investimento
- [x] Cálculo automático de rentabilidade
- [x] Gráficos de performance
- [x] Tipos diversos (ações, cripto, imóveis, etc)

## 6. Metas Financeiras

- [x] Criação de metas
- [x] Acompanhamento de progresso
- [x] Edição de metas
- [x] Exclusão de metas
- [x] Marcação de meta concluída

## 7. Análises com IA

- [x] Geração de análise mensal
- [x] Insights personalizados
- [x] Recomendações de economia
- [x] Histórico de análises

## 8. Admin (Painel Administrativo)

- [x] Listagem de usuários
- [x] Filtros (status, role)
- [x] Edição de usuário
- [x] Ativação/desativação de usuário
- [x] Reenvio de email de boas-vindas
- [x] Visualização de detalhes

## 9. Integração Hotmart

- [x] Webhook para eventos (compra, cancelamento)
- [x] Ativação automática de usuário
- [x] Desativação automática em cancelamento
- [x] Monitoramento de webhook
- [x] Link de checkout configurado



## DESIGN E UX

---

### Paleta de Cores

- **Primary:** #00bf63 (Verde) - Receitas e CTAs
- **Danger:** #ef4444 (Vermelho) - Despesas e ações destrutivas
- **Warning:** #f59e0b (Amarelo) - Investimentos e alertas
- **Neutral:** #737373 (Cinza) - Elementos secundários
- **Background:** #f9fafb (Cinza claro) - Fundo da aplicação



## Componentes UI

- Buttons (Primary, Secondary, Outline, Ghost, Link, Destructive)
- Cards (com hover e shadow)
- Modals/Dialogs (centro da tela)
- Drawers (lateral)
- Toasts (notificações)
- Tables (responsivas)
- Forms (validação em tempo real)
- Charts (interativos)
- Loading states (spinners)
- Empty states (ilustrações)

## Responsividade

- ✓ Desktop (1920px+)
- ✓ Laptop (1280px-1919px)
- ✓ Tablet (768px-1279px)
- ✓ Mobile (320px-767px)



## PROBLEMAS RESOLVIDOS

### 1. Erro: Value 'INVESTMENT' not found in enum

**Causa:** Prisma Client gerado antes da migração

**Solução:** Trocar ordem no vercel.json (migrate → generate → build)

**Documentos:** SOLUCAO\_INVESTMENT\_ENUM.md , CORRECAO\_ORDEM\_PRISMA\_VERCEL.md

### 2. Erro: Authentication failed (Supabase)

**Causa:** DATABASE\_URL apontando para Supabase em vez de Abacus

**Solução:** Atualizar variável no Vercel com URL completa

**Documentos:** PASSO\_FINAL\_VERCEL.md , CONFIRMACAO\_FINAL\_BANCO\_ABACUS.md

### 3. Erro: pgbouncer not configured

**Causa:** Falta parâmetro pgbouncer=true na URL

**Solução:** Adicionar parâmetros de pooling à DATABASE\_URL

**Documentos:** SOLUCAO\_DEFINITIVA\_DATABASE\_URL.md

### 4. Modal LGPD não aparecendo

**Causa:** Flag firstLogin não estava sendo verificada

**Solução:** Adicionar verificação e modal no dashboard

**Documentos:** INTEGRACAO\_LGPD\_FIRST\_LOGIN.md

### 5. Build timeout no Vercel

**Causa:** Conexão direta ao banco (não pooled)

**Solução:** Usar pgbouncer para connection pooling

**Documentos:** CORRECAO\_TIMEOUT\_DB.md

---

## **ESTRUTURA DE ARQUIVOS**

---

```

/home/ubuntu/orcamento_planejado/
├── nextjs_space/
│   └── app/
│       ├── api/
│       │   ├── admin/
│       │   ├── analyses/
│       │   ├── auth/
│       │   ├── categories/
│       │   ├── goals/
│       │   ├── investments/
│       │   ├── profile/
│       │   ├── signup/
│       │   ├── transactions/
│       │   ├── user/
│       │   └── webhook/
│       ├── auth/
│       │   ├── login/
│       │   ├── register/
│       │   ├── forgot-password/
│       │   └── reset-password/
│       ├── dashboard/
│       │   ├── admin/
│       │   ├── analyses/
│       │   ├── categories/
│       │   ├── first-login/
│       │   ├── investments/
│       │   ├── transactions/
│       │   ├── layout.tsx
│       │   └── page.tsx
│       ├── globals.css
│       ├── layout.tsx
│       └── page.tsx
│   └── components/
│       ├── admin/
│       ├── analyses/
│       ├── auth/
│       ├── categories/
│       ├── dashboard/
│       ├── investments/
│       ├── navigation/
│       ├── profile/
│       ├── providers/
│       ├── transactions/
│       └── ui/
│   └── lib/
│       ├── auth.ts
│       ├── db.ts
│       ├── email.ts
│       ├── prisma-helpers.ts
│       ├── types.ts
│       └── utils.ts
│   └── prisma/
│       ├── migrations/
│       └── schema.prisma
│   └── scripts/
│       ├── check-db-url.js
│       └── seed.ts
│   ├── .env
│   ├── middleware.ts
│   ├── next.config.js
│   ├── package.json
│   └── tailwind.config.ts

```

```

# API Routes
# Endpoints admin
# Análises IA
# Autenticação
# Categorias
# Metas
# Investimentos
# Perfil
# Registro
# Transações
# Usuário
# Webhooks Hotmart
# Páginas de autenticação

# Páginas do dashboard
# Painel admin
# Análises
# Categorias
# Primeiro login
# Investimentos
# Transações
# Layout do dashboard
# Página principal
# Estilos globais
# Layout raiz
# Landing page














# Componentes admin
# Componentes análises
# Componentes autenticação
# Componentes categorias
# Componentes dashboard
# Componentes investimentos
# Navegação
# Perfil
# Providers (Auth, Theme)
# Componentes transações
# Componentes UI base

# Configuração NextAuth
# Cliente Prisma
# Envio de emails
# Helpers Prisma
# Tipos TypeScript
# Utilitários

# Migrações
# Schema do banco

# Valida DATABASE_URL
# Seed inicial
# Variáveis de ambiente
# Middleware Next.js
# Config Next.js
# Dependências
# Config Tailwind

```

		tsconfig.json		Config TypeScript
		vercel.json		Config Vercel
		[DOCUMENTAÇÃO]		
		SOLUCAO_INVESTMENT_ENUM.md		
		CORRECAO_ORDEM_PRISMA_VERCEL.md		
		PASSO_FINAL_VERCEL.md		
		CONFIRMACAO_FINAL_BANCO_ABACUS.md		
		STATUS_FINAL_CORRECOES.md		
		RELATORIO_COMPLETO_PROJETO.md (este arquivo)		

## SEGURANÇA

### Implementado

- ✓ Hashing de senhas (bcrypt)
- ✓ JWT para sessões (NEXTAUTH\_SECRET)
- ✓ HTTPS em produção (Vercel)
- ✓ Middleware de autenticação
- ✓ Validação de input (Zod)
- ✓ Proteção CSRF (NextAuth)
- ✓ Rate limiting (Vercel automático)
- ✓ Environment variables isoladas
- ✓ Webhook secret (Hotmart)
- ✓ Consentimento LGPD

### Recomendações Futuras

- ☐ Rate limiting customizado (em rotas críticas)
- ☐ 2FA (autenticação de dois fatores)
- ☐ Logs de auditoria
- ☐ Monitoramento de atividades suspeitas
- ☐ Backup automático do banco
- ☐ Criptografia de dados sensíveis

## MÉTRICAS E MONITORAMENTO

### Implementado

- ✓ Logs de erro no console
- ✓ Logs de webhook Hotmart
- ✓ Verificação de DATABASE\_URL no build

### Recomendações Futuras

- ☐ Sentry (error tracking)
- ☐ Google Analytics (comportamento)
- ☐ Vercel Analytics (performance)
- ☐ Posthog (product analytics)
- ☐ Logs estruturados (Winston/Pino)

---

## COMANDOS IMPORTANTES

---

### Desenvolvimento Local

```
cd /home/ubuntu/orcamento_planejado/nextjs_space

# Iniciar servidor de desenvolvimento
yarn dev

# Verificar Prisma Client
node -e "const {TransactionType} = require('@prisma/client'); console.log(TransactionType);"

# Gerar Prisma Client
yarn prisma generate

# Aplicar migrações
yarn prisma migrate deploy

# Build de produção
yarn build

# Iniciar servidor de produção
yarn start
```

### Prisma

```
# Criar migração
npx prisma migrate dev --name nome_da_migracao

# Visualizar banco de dados
npx prisma studio

# Reset do banco (CUIDADO!)
npx prisma migrate reset

# Seed do banco
npx prisma db seed
```

### Git

```
# Status
git status

# Add all
git add -A

# Commit
git commit -m "mensagem"

# Push
git push origin main
```

---

## PRÓXIMOS PASSOS (Roadmap)

---

### Curto Prazo (1-2 semanas)

- [ ] Testes E2E (Playwright/Cypress)
- [ ] Testes unitários (Jest/Vitest)
- [ ] Melhorias de UX (loading states, animações)
- [ ] Relatórios em PDF
- [ ] Exportação de dados completa

### Médio Prazo (1-2 meses)

- [ ] App Android (React Native/PWA)
- [ ] Notificações push
- [ ] Sistema de alertas (metas, limites)
- [ ] Dashboard personalizado
- [ ] Mais tipos de gráficos

### Longo Prazo (3-6 meses)

- [ ] Planos diferentes (Basic, Pro, Premium)
- [ ] Integração bancária (Open Banking)
- [ ] Compartilhamento de orçamento familiar
- [ ] IA mais avançada (previsões, recomendações)
- [ ] Marketplace de consultores

---

## CONTATOS E SUPORTE

---

**Desenvolvedor:** Vinicius Leber

**Email:** [viniciusleber@gmail.com](mailto:viniciusleber@gmail.com)

**Email de Suporte:** [suporteplanejado@gmail.com](mailto:suporteplanejado@gmail.com)

**GitHub:** <https://github.com/lebervinicius-dev>




---

## NOTAS IMPORTANTES PARA O DEEPAGENT




---

### Ao Continuar Este Projeto


#### 1. Banco de Dados:

-  SEMPRE usar a DATABASE\_URL com `pgbouncer=true`
-  Verificar que migrações foram aplicadas antes de generate
-  Nunca fazer reset em produção

#### 2. Enums:

-  TransactionType tem 3 valores: INCOME, EXPENSE, INVESTMENT
-  CategoryType tem 3 valores: INCOME, EXPENSE, INVESTMENT
-  Sempre regenerar Prisma Client após mudanças no schema

#### 3. Deploy:

-  Ordem no vercel.json: migrate → generate → build

- ☒ Todas as variáveis devem estar no Vercel
- ☒ .env não deve ser commitado (está no .gitignore)

#### 4. Testes:

- ☒ Usuário teste: teste@teste.com / teste123
- ☒ Admin teste: admin@teste.com / admin123
- ☒ Sempre testar fluxo completo após mudanças

#### 5. Documentação:

- ☒ Criar .md para cada problema/solução importante
- ☒ Manter este relatório atualizado
- ☒ Gerar PDF automaticamente

---

## ☒ CHECKLIST DE VERIFICAÇÃO

---

Antes de fazer qualquer alteração:

- ☐ Backup do banco (exportar dados importantes)
- ☐ Branch do git atualizada
- ☐ Variáveis de ambiente verificadas
- ☐ Prisma Client atualizado
- ☐ Build local funcionando
- ☐ Testes manuais realizados

Após fazer alterações:

- ☐ Yarn build executado com sucesso
  - ☐ Testes manuais aprovados
  - ☐ Commit e push realizados
  - ☐ Vercel deployment bem-sucedido
  - ☐ Testes em produção aprovados
  - ☐ Documentação atualizada
- 

## CONCLUSÃO

---

Este projeto está **100% funcional e em produção**.

#### Status atual:

- ☒ Código estável e testado
- ☒ Banco de dados configurado corretamente
- ☒ Deploy automatizado funcionando
- ☒ Todas as funcionalidades principais implementadas
- ☒ Documentação completa e atualizada

#### Para novas conversas:

- Use este documento como referência principal
- Leia os documentos específicos para problemas conhecidos

- Sempre verifique o status do Prisma Client antes de mudanças
- Mantenha a ordem de comandos no vercel.json

**Em caso de dúvidas:**

- Consulte os arquivos .md na raiz do projeto
- Verifique os logs do Vercel
- Execute comandos de verificação localmente
- Sempre teste antes de fazer push

---

**Última atualização:** 2025-11-01 04:10 UTC

**Versão:** 2.0 - Relatório Completo

**Autor:** DeepAgent + Vinicius Leber

**Status:**  Produção