



FASE 1 - FUNDAÇÃO COMPLETA

Resumo da Migração

Migração bem-sucedida de funcionalidades críticas do **Professor Dindin** para o **Dindin Teens**, adaptando para o público jovem (12-19 anos) com linguagem teen, emojis e gamificação!



Sprint 1.1: Schema Prisma Completo

Modelos Criados/Migrados (30+ modelos)

Autenticação & Usuários

- `User` - Usuários com roles: TEEN, PROFESSOR, RESPONSIBLE, ADMIN
- `Account` , `Session` , `VerificationToken` - NextAuth
- `TeenProfile` - Perfil específico para adolescentes
- `UserProgress` - XP, níveis, streaks

Testes & Avaliações

- `CheckupTest` - Check-up de Estresse Financeiro
- `RaioXTest` - Raio-X da Personalidade Financeira
- `MapaTesouroTest` - Mapa do Tesouro com ISJF completo

Sistema ISJF (BRAVO360)

- `ControlObjective` - 22 Objetivos de Controle
- `ControlAssessment` - Avaliações dos objetivos
- `ISJFHistory` - Histórico de ISJFs calculados

Jornada Financeira

- `JornadaFinanceira` - Status geral da jornada
- `EventoJornada` - Timeline de eventos e marcos






Gamificação

- `Activity` - 30 atividades do banco
- `CompletedActivity` - Atividades completadas
- `AtividadeProgresso` - Progresso detalhado
- `Badge` - Badges de conquista com Core Drives
- `UserBadge` - Badges conquistadas
- `DailyMission` - Missões diárias
- `CompletedMission` - Missões completadas
- `Streak` - Sequências de dias ativos



Agentes Estressores

- `StressorAgent` - Agentes de estresse financeiro
- `StressorAssessment` - Avaliações de estresse



Instituições & Turmas

-  EducationalInstitution - Escolas/faculdades
-  Class - Turmas
-  ProfessorClass - Professores vinculados
-  TeenClass - Alunos matriculados
-  TeenResponsible - Responsáveis dos teens





Mensageria & Notificações

-  Message - Sistema de mensagens
-  Notificacao - Notificações e alertas

Auditoria & Segurança





-  AuditLog - Log de auditoria
-  NewsletterSubscriber - Newsletter

Enums Adicionados

-  UserRole : TEEN, PROFESSOR, RESPONSIBLE, ADMIN
-  IncomeSource : MESADA, TRABALHO, FREELANCE, BICO, NENHUM
-  MissionType : COMPLETE_ACTIVITIES, EARN_XP, MAINTAIN_STREAK, UNLOCK_BADGES, TAKE_TESTS
-  Outros: Archetype, RiskProfile, ActivityModule, CoreDrive

Sprint 1.2: Seed Completo

Dados Populados

-  **4 Usuários de Teste**
 - Admin (admin@dindinteens.com.br / admin123)
 - Teen (lucas@teste.com / teen123)
 - Professor (maria@escola.com / prof123)
 - Responsável (ana@teste.com / resp123)
-  **22 Objetivos de Controle ISJF**
 - Todos com categoria, peso e ordem
 - Descrições adaptadas para linguagem teen
 - Exemplos práticos para adolescentes
-  **30 Atividades do Banco**
 - Checkup (CK-01 a CK-10)
 - Raio-X (RX-01 a RX-10)
 - Mapa Tesouro (MT-01 a MT-10)
 - Todas com linguagem teen e emojis
-  **10 Badges de Gamificação**
 - Associadas aos Core Drives do Octalysis

- Critérios claros de conquista
- **✓ 8 Agentes Estressores**
- Adaptados para realidade teen
- Com exemplos e dicas
- **✓ 3 Missões Diárias**
- Tipos variados de desafios
- Sistema de recompensas
- **✓ 1 Instituição + 1 Turma**
- Professor e teen vinculados
- Responsável associado ao teen

Scripts Configurados

```
{
  "db:seed": "ts-node prisma/seed.ts",
  "db:migrate": "prisma migrate dev",
  "db:reset": "prisma migrate reset",
  "db:generate": "prisma generate"
}
```

✓ Sprint 1.3: Motor ISJF (BRAVO360)

Arquivos Criados

/src/lib/escalas-isjf.ts

- ✓ Escalas de conversão adaptadas para teens
- ✓ Funções: `converterImportancia`, `converterDificuldade`, `converterFrequencia`
- ✓ Textos descritivos com emojis

/src/lib/classificacao-isjf.ts

- ✓ 5 Classificações: Fragilidade Crítica/Alta, Resiliente, Robusto, Antifrágil
- ✓ Funções de classificação de objetivos (RESTRICTOR/FACILITADOR)
- ✓ Mensagens contextuais em linguagem teen
- ✓ Classes Tailwind para UI

/src/lib/mapeamento-objetivos-isjf.ts

- ✓ 22 Objetivos de Controle com siglas (ACC, BS, CD, etc.)
- ✓ Mapeamento para 4 Determinantes (GAR, HAB, REC, RI)
- ✓ Variáveis compostas (INF_GERAL, SEG_GERAL)
- ✓ Funções auxiliares (getObjetivoBySigla, etc.)

/src/lib/calculo-isjf.ts (ARQUIVO PRINCIPAL)

- ✓ Constantes ISJF (Kf=711, Ks=2036, Ki=125, Kg=10, Kj=5)
- ✓ Cálculo IER: $(125 - X*Y*Z) / 10$
- ✓ Cálculo IRB360: $((X*Y)*Z) * 5$

- ✓ 4 Variáveis Comportamentais (M1, M2, M3, M4)
- ✓ 6 Determinantes Executivos (GAR, HAB, REC, RI, OP, UTIL)
- ✓ Fórmula Completa ISJF: $((((\text{MÉDIA}(\text{GAR}, \text{HAB}, \text{REC})) \times (\text{RI} / \text{Kf})) / \text{Ks}) \times \text{OP}) ^ \text{UTIL}$
- ✓ Separação de Restrictores e Facilitadores
- ✓ Funções de formatação

APIs REST Criadas

POST /api/isjf/calculate

```
// Calcula ISJF completo das 22 respostas
Body: {
  userId?: string,
  respostas: [{ id, importancia, dificuldade, frequencia }, ...] // 22 respostas
}
Response: {
  isjf, classificacao, determinantes, variaveis,
  restrictores, facilitadores, historyId
}
```

GET /api/isjf/latest?userId=xxx

```
// Retorna último ISJF calculado
Response: { data: ISJFHistory }
```

GET /api/isjf/history?userId=xxx&limit=10

```
// Retorna histórico de ISJFs
Response: { data: ISJFHistory[], total, limit }
```

✓ Sprint 1.4: Motor de Recomendações (IA)

Arquivo Criado: /src/lib/motor-recomendacoes.ts

Funcionalidades

- ✓ Integração com Abacus.AI via fetch API
- ✓ Prompts contextuais adaptados para teens
- ✓ Sistema de fallback inteligente
- ✓ Referências aos 7 modelos teóricos:
 1. **Taleb** - Antifragilidade
 2. **Falconi** - PDCA
 3. **Freud** - Id, Ego, Superego
 4. **Piaget** - Desenvolvimento
 5. **Kurt Lewin** - Mudança
 6. **Kübler-Ross** - Aceitação
 7. **BRAVO360** - Performance

Estrutura de Recomendação

```
{
  titulo: string,
  mensagem: string,           // Motivacional, max 200 palavras
  acaoImediata: string[],     // 3-5 ações AGORA
  proximosPassos: string[],   // 3-5 passos médio prazo
  atividadesSugeridas: string[], // Códigos CK-XX, RX-XX, MT-XX
  motivacao: string,          // Frase inspiradora
  referenciasModelos: string[] // 2-3 modelos teóricos
}
```

API Criada

POST /api/recomendacoes

```
Body: {
  userId?: string,
  modulo: 'checkup' | 'raio-x' | 'mapa-tesouro',
  dados: {
    respostas?: [...],           // Para mapa-tesouro
    agenteEstressor?: string,    // Para checkup
    perfilPsicoFinanceiro?: string // Para raio-x
  }
}
Response: { recomendacao: RecomendacaoGerada }
```



Configuração Necessária

1. Banco de Dados PostgreSQL

```
# Atualizar DATABASE_URL no .env
DATABASE_URL="postgresql://user:password@localhost:5432/dindin_teens"
```

2. Executar Migrações

```
npm run db:migrate
```

3. Popular Banco (Seed)

```
npm run db:seed
```

4. Configurar Abacus.AI (Opcional)

```
# Adicionar no .env
ABACUS_API_KEY="your-abacus-api-key"
```

5. NextAuth Secret

```
# Gerar secret
openssl rand -base64 32

# Adicionar no .env
NEXTAUTH_SECRET="generated-secret"
```



Estatísticas da Migração

- **Modelos Prisma:** 30+
- **Enums:** 8
- **Arquivos Lib:** 5 (ISJF) + 1 (Recomendações)
- **APIs REST:** 7 endpoints
- **Linhas de Código:** ~3.000+
- **Objetivos ISJF:** 22
- **Atividades:** 30
- **Badges:** 10
- **Usuários Teste:** 4



Próximas Fases (Roadmap)

FASE 2 - INTERFACES & UX

- [] Páginas de Testes (Checkup, Raio-X, Mapa)
- [] Dashboard do Teen
- [] Perfil e Configurações
- [] Timeline da Jornada
- [] Sistema de Badges visual

FASE 3 - GAMIFICAÇÃO AVANÇADA

- [] Sistema de XP e Níveis
- [] Missões Diárias dinâmicas
- [] Leaderboards (opcional)
- [] Sistema de Streaks visual

FASE 4 - FUNCIONALIDADES SOCIAIS

- [] Sistema de Mensagens
- [] Notificações Push
- [] Dashboard do Professor
- [] Dashboard do Responsável
- [] Relatórios e Analytics

FASE 5 - REFINAMENTO & DEPLOY

- [] Testes E2E

- [] Otimização de Performance
- [] Deploy Production
- [] Monitoramento e Logs
- [] Documentação API completa



Problemas Conhecidos & Soluções

1. Prisma Client não encontrado

Solução: `npm run db:generate`

2. Erro de conexão DB

Solução: Configurar DATABASE_URL no .env

3. Seed falha

Solução: Verificar se DB está rodando e executar migrações primeiro

4. APIs retornam 401

Solução: Autenticar usuário ou passar userId no body



Notas de Desenvolvimento

Adaptações para Teens

- ☒ Linguagem casual com emojis
- ☒ Exemplos relevantes (mesada, jogos, streaming)
- ☒ Valores ajustados (R\$ 5-500 vs R\$ 500-5000)
- ☒ Tom motivacional e sem julgamento
- ☒ Interface gamificada

Diferenças do Professor Dindin

- Schema simplificado (removido campos legado)
- APIs RESTful (vs server components)
- TypeScript strict mode
- Motor IA com fallback robusto
- Foco em mobile-first



Conclusão

A **FASE 1 - FUNDAÇÃO COMPLETA** foi executada com sucesso! ☒

O Dindin Teens agora possui:

- ☒ Schema Prisma robusto e completo
- ☒ Sistema ISJF (BRAVO360) 100% funcional
- ☒ Motor de Recomendações com IA

- ☒ Seed completo com dados de teste
- ☒ APIs REST testadas e funcionais
- ☒ Linguagem adaptada para público jovem

Pronto para FASE 2! 🚀

Data de Conclusão: 31 de Janeiro de 2026

Status: ☒ COMPLETO

Próximo Passo: Iniciar FASE 2 - Interfaces & UX