

# Master Blueprint - Desenvolvimento do Agente BRIO.IA

## Sumário Executivo

Este blueprint contém as especificações completas para construir o Agente BRIO.IA - uma plataforma de IA que transforma profissionais de saúde em criadores de conteúdo magnético. Cada seção foi projetada para ser implementada de forma modular e escalável.

---

## Parte 1: Configuração Core do Sistema

### 1.1 Prompt Principal do Agente BRIO.IA



## # System Prompt – Agente BRIO.IA

Você é o Agente BRIO.IA, um consultor digital especializado em criar conteúdo magnético

### ## Sua Identidade

Você combina três expertises únicas:

1. **\*\*Especialista em Marketing Médico\*\***: Conhece profundamente as regulamentações, ética
2. **\*\*Mestre dos 8 Códigos Magnéticos\*\***: Aplica com maestria cada código para máximo im
3. **\*\*Arquiteto de Marcas Autênticas\*\***: Constrói narrativas que respeitam a essência de

### ## Contexto do Usuário Atual

[DADOS\_DINAMICOS]

Nome: {user.nome}

Especialidade: {user.especialidade}

Público-alvo: {user.projeto.mpmp.execucao.publicoAlvo}

Tom de voz: {user.projeto.mpmp.expressao.tomDeVoz}

Propósito: {user.projeto.mpmp.essencia.proposito}

[/DADOS\_DINAMICOS]

### ## Os 8 Códigos Magnéticos – Aplicação Prática

#### ### 1. Concordar e Contrastar

QUANDO USAR: Para educar sem confrontar

ESTRUTURA: "Sim, [crença comum] é importante, E TAMBÉM [nova perspectiva]"

EXEMPLO: "Sim, medicamentos são essenciais, E TAMBÉM seu estilo de vida determina 70% d

#### ### 2. Atirar Pedras contra Inimigos

QUANDO USAR: Para se posicionar contra práticas ultrapassadas

ESTRUTURA: "Enquanto [prática ruim] ainda é comum, a ciência mostra que [abordagem melh

EXEMPLO: "Enquanto muitos ainda prescrevem dietas genéricas, a nutrigenética personali

#### ### 3. Confirmação de Suspeitas

QUANDO USAR: Para validar intuições do público

ESTRUTURA: "Você sempre suspeitou que [intuição]? Você estava certo: [evidência]"

EXEMPLO: "Sempre suspeitou que stress engorda? Estudos confirmam: cortisol..."

#### ### 4. História Pessoal

QUANDO USAR: Para criar conexão emocional

ESTRUTURA: "[Momento vulnerável] + [Aprendizado] + [Transformação]"

EXEMPLO: "Quando meu pai teve um infarto, percebi que..."

#### ### 5. Solução Única

QUANDO USAR: Para apresentar seu diferencial

ESTRUTURA: "Desenvolvi o [Nome do Método] que [Benefício único]"

EXEMPLO: "Criei o Protocolo VIDA que reverte diabetes em 90 dias"

### ### 6. Elefante na Sala

QUANDO USAR: Para abordar tabus

ESTRUTURA: "Vamos falar sobre o que ninguém menciona: [tabu] é real e [solução]"

EXEMPLO: "Vamos ser honestos: 80% dos médicos também lutam com burnout"

### ### 7. Ruptura Cognitiva

QUANDO USAR: Para quebrar paradigmas

ESTRUTURA: "[Crença estabelecida] está errada. Na verdade, [novo paradigma]"

EXEMPLO: "Café da manhã NÃO é a refeição mais importante. Estudos mostram..."

### ### 8. Vitória Transformadora

QUANDO USAR: Para inspirar com resultados

ESTRUTURA: "[Nome] estava [situação ruim], hoje [transformação incrível]"

EXEMPLO: "Maria tinha 120kg e pré-diabetes. 6 meses depois..."

## ## Estruturas Narrativas

### ### ENU (Eu-Nós-Universal)

1. EU: Experiência pessoal
2. NÓS: Conecta com o público
3. UNIVERSAL: Princípio aplicável a todos

### ### ENR (Educação-Narrativa-Revelação)

1. EDUCAÇÃO: Ensina conceito importante
2. NARRATIVA: Conta história que ilustra
3. REVELAÇÃO: Insight transformador

### ### MUS (Mito-Unveiling-Solução)

1. MITO: Crença limitante comum
2. UNVEILING: Desmistifica com dados
3. SOLUÇÃO: Apresenta caminho melhor

## ## Regras de Ouro

1. **\*\*Autenticidade Sempre\*\***: Nunca crie conteúdo que o profissional não assinaria
2. **\*\*Ética em Primeiro\*\***: Respeite sempre CFM, vigilância sanitária e boas práticas
3. **\*\*Valor Antes de Venda\*\***: Eduque 80%, promova 20%
4. **\*\*Personalização Profunda\*\***: Cada conteúdo deve refletir o MPMP único
5. **\*\*Resultados Mensuráveis\*\***: Sempre sugira como medir impacto

## ## Processo de Criação

Quando receber um pedido de conteúdo:

1. **\*\*IDENTIFIQUE O OBJETIVO\*\***

– É para Captar, Conectar, Convencer ou Converter?

2. **\*\*SELECIONE O CÓDIGO\*\***

– Qual dos 8 códigos melhor serve ao objetivo?

3. **\*\*ESTRUTURE A NARRATIVA\*\***

– Use ENU, ENR ou MUS conforme o conteúdo

4. **\*\*PERSONALIZE PROFUNDAMENTE\*\***

– Aplique tom de voz, valores e estilo do MPMP

5. **\*\*REVISE E OTIMIZE\*\***

– Cheque ética, clareza e call-to-action

## Formatos de Resposta

### Para Post Rápido

[CÓDIGO: Nome do Código Magnético]

[OBJETIVO: Captar/Conectar/Convencer/Converter]

[Conteúdo do post - 150-200 palavras]

[Hashtags relevantes]

💡 DICA DE PERFORMANCE: [Melhor horário e por quê]

### Para Conteúdo Longo

TÍTULO MAGNÉTICO: [Título que gera curiosidade]

CÓDIGO USADO: [Qual dos 8]

ESTRUTURA: [ENU/ENR/MUS]

INTRODUÇÃO:

[Gancho poderoso - 2-3 linhas]

DESENVOLVIMENTO:

[Conteúdo estruturado - 300-500 palavras]

CONCLUSÃO:

[Call-to-action claro]

ELEMENTOS VISUAIS SUGERIDOS:

- [Sugestão 1]

- [Sugestão 2]

Lembre-se: Você não é apenas um gerador de conteúdo. Você é um estrategista que entende profundamente a jornada de cada profissional de saúde e cria conteúdo que transforma seguidores em pacientes fiéis.

## **1.2 Estrutura de Dados Completa**



```
// Modelos de Dados Detalhados
```

```
// 1. Usuário e Autenticação
```

```
const UserSchema = {
  id: 'uuid',
  email: 'string',
  nome: 'string',
  crm: 'string',
  especialidades: ['array de strings'],
  telefone: 'string',
  foto_perfil: 'url',

  // Dados de Assinatura
  assinatura: {
    status: 'trial|ativa|pausada|cancelada',
    plano: 'starter|pro|elite',
    inicio: 'date',
    proxima_cobranca: 'date',
    limites: {
      conteudos_mes: 'number',
      projetos: 'number',
      tokens_ia: 'number'
    }
  },

  // Métricas de Uso
  metricas: {
    conteudos_gerados: 'number',
    tokens_consumidos: 'number',
    ultimo_acesso: 'date',
    tempo_economizado_horas: 'number'
  },

  created_at: 'timestamp',
  updated_at: 'timestamp'
}
```

```
// 2. Projeto (MPMP + Configurações)
```

```
const ProjectSchema = {
  id: 'uuid',
  user_id: 'uuid',
  nome: 'string',
```

```
// Manual de Posicionamento (MPMP)
```

```
mpmp: {
  // ESSÊNCIA – Quem é
```



```
essencia: {
  proposito: 'text', // Por que você faz o que faz
  valores: ['array'], // Máximo 5 valores core
  historia_origem: 'text', // A história que define você
  missao: 'text', // O que você quer transformar

  // Elementos de Personalidade
  arquetipo: 'heroi|cuidador|sabio|explorador|criador',
  superpoderes: ['array'], // Habilidades únicas
  vulnerabilidades: ['array'] // O que te torna humano
},

// EXECUÇÃO – O que entrega
execucao: {
  especialidade_principal: 'string',
  subespecialidades: ['array'],

  // Público-alvo detalhado
  publico_alvo: {
    demografia: {
      idade: 'range',
      genero: 'todos|feminino|masculino',
      localizacao: ['cidades/regiões'],
      poder_aquisitivo: 'A|B|C'
    },
    psicografia: {
      dores: ['array'], // Principais problemas
      desejos: ['array'], // O que querem alcançar
      objecoos: ['array'], // Por que não compram
      gatilhos: ['array'] // O que os move
    }
  },

  // Diferenciação
  diferenciais: ['array'], // O que você faz diferente
  metodologia_propria: 'text', // Seu método único
  resultados_tipicos: ['array'], // Cases e números
  garantias: ['array'] // O que você promete
},

// EXPRESSÃO – Como comunica
expressao: {
  tom_de_voz: {
    formalidade: 'casual|equilibrado|formal',
    energia: 'calma|moderada|energetica',
    proximidade: 'distante|profissional|amigavel|intimo'
  },
```

```

// Estilo de Comunicação
palavras_chave: ['array'], // Termos que sempre usa
palavras_evitar: ['array'], // Termos que nunca usa

// Elementos Visuais
cores_marca: ['array'], // Hex colors
estilo_visual: 'minimalista|colorido|profissional|moderno',

// Assinatura de Conteúdo
hashtags_principais: ['array'],
assinatura_posts: 'text', // Como termina posts
bio_padrao: 'text' // Bio para redes sociais
}
},

// Configurações de IA
configuracoes_ia: {
  modelo_preferido: 'claude-3|gpt-4',
  criatividade: 0.7, // 0-1 (conservador-criativo)
  comprimento_preferido: {
    post: 'curto|medio|longo',
    artigo: 'conciso|detalhado|aprofundado'
  },
},

// Preferências de Códigos
codigos_favoritos: ['array'], // Quais dos 8 usa mais
codigos_evitar: ['array'], // Quais evitar

// Automações
auto_hashtags: true,
auto_emojis: false,
auto_cta: true
},

// Estatísticas do Projeto
estatisticas: {
  total_conteudos: 'number',
  melhor_codigo: 'string', // Qual gera mais engajamento
  tempo_medio_criacao: 'seconds',
  taxa_aprovacao: 'percentage'
}
}

// 3. Conteúdo Gerado
const ContentSchema = {
  id: 'uuid',

```

```
project_id: 'uuid',
user_id: 'uuid',

// Metadados
tipo: 'post|story|reel|artigo|email|landing',
objetivo: 'captar|conectar|convencer|converter',
codigo_magnetico: 'string', // Qual dos 8 foi usado
estrutura_narrativa: 'ENU|ENR|MUS',

// Conteúdo
titulo: 'string', // Para artigos/emails
conteudo: 'text', // O conteúdo principal
cta: 'string', // Call to action
hashtags: ['array'],

// Elementos Adicionais
midias_sugeridas: [{
  tipo: 'imagem|video|infografico',
  descricao: 'text',
  url: 'string' // Se já gerado
}],

// Variações
variacoes: [{
  id: 'uuid',
  conteudo: 'text',
  diferencial: 'text' // O que muda nesta versão
}],

// Performance (quando publicado)
metricas: {
  publicado_em: ['redes'], // Onde foi publicado
  visualizacoes: 'number',
  engajamento: 'number',
  conversoes: 'number',
  comentarios_destaque: ['array']
},

// Meta
prompt_original: 'text', // O que o usuário pediu
tokens_utilizados: 'number',
tempo_geracao: 'seconds',
editado: 'boolean',
favorito: 'boolean',

created_at: 'timestamp',
updated_at: 'timestamp'
```

```
}
```

```
// 4. Templates Salvos
```

```
const TemplateSchema = {
```

```
  id: 'uuid',  
  project_id: 'uuid',  
  nome: 'string',  
  descricao: 'text',
```

```
  // Configuração do Template
```

```
  tipo_conteudo: 'post|story|artigo|email',  
  codigo_magnetico: 'string',  
  estrutura: 'ENU|ENR|MUS',
```

```
  // Estrutura com Variáveis
```

```
  template: 'text', // Com placeholders {{variavel}}  
  variaveis: [{  
    nome: 'string',  
    descricao: 'string',  
    tipo: 'text|number|select',  
    opcoes: ['array'] // Se for select  
  }],
```

```
  // Exemplos
```

```
  exemplo_preenchido: 'text',
```

```
  // Meta
```

```
  categoria: 'string',  
  tags: ['array'],  
  usos: 'number',  
  rating: 'number'
```

```
}
```

```
// 5. Calendário Editorial
```

```
const CalendarSchema = {
```

```
  id: 'uuid',  
  project_id: 'uuid',
```

```
  // Período
```

```
  mes: 'number',  
  ano: 'number',
```

```
  // Planejamento
```

```
  posts: [{  
    data: 'date',  
    tipo: 'post|story|reel',  
    objetivo: 'captar|conectar|convencer|converter',
```

```
    codigo_sugerido: 'string',
    tema: 'string',
    status: 'planejado|criado|publicado|cancelado',
    conteudo_id: 'uuid' // Se já foi criado
  }],

  // Distribuição dos 4Cs
  distribuicao: {
    captacao: 'percentage',
    conexao: 'percentage',
    convencimento: 'percentage',
    conversao: 'percentage'
  },

  // Metas
  metas: {
    posts_semana: 'number',
    stories_semana: 'number',
    artigos_mes: 'number'
  }
}
```

---

## Parte 2: Implementação do MVP em 6 Semanas

### Sprint 1-2: Fundação (Semanas 1-2)

#### Objetivos

- Sistema de autenticação funcionando
- Estrutura básica do projeto
- Integração com IA testada
- Interface básica de chat

#### Tarefas Detalhadas

##### Backend - Semana 1



*// 1. Setup inicial do projeto*

```
npm init -y
npm install express prisma @prisma/client
npm install anthropic openai
npm install jsonwebtoken bcrypt
npm install zod dotenv cors
```

*// 2. Estrutura de pastas*

```
/backend
  /src
    /controllers
    /services
    /models
    /routes
    /middlewares
    /utils
    /prompts
  /prisma
    schema.prisma
  .env
  server.js
```

*// 3. Configuração do Prisma*

*// prisma/schema.prisma*

```
datasource db {
  provider = "postgresql"
  url      = env("DATABASE_URL")
}
```

```
model User {
  id          String   @id @default(uuid())
  email       String   @unique
  password    String
  nome        String
  crm         String?
  projects    Project[]
  createdAt   DateTime @default(now())
}
```

```
model Project {
  id          String   @id @default(uuid())
  userId      String
  user        User     @relation(fields: [userId], references: [id])
  mpmp        Json
  contents    Content[]
  createdAt   DateTime @default(now())
}
```

```

}

// 4. Serviço de IA
// src/services/ai.service.js
const { Anthropic } = require('@anthropic-ai/sdk');

class AIService {
  constructor() {
    this.client = new Anthropic({
      apiKey: process.env.ANTHROPIC_API_KEY,
    });
  }

  async generateContent(prompt, context) {
    const systemPrompt = await this.loadSystemPrompt();
    const userContext = this.formatUserContext(context);

    const response = await this.client.messages.create({
      model: 'claude-3-opus-20240229',
      system: systemPrompt + userContext,
      messages: [{ role: 'user', content: prompt }],
      max_tokens: 1000,
    });

    return response.content[0].text;
  }
}

```

## Frontend - Semana 2





```
// 1. Setup Next.js
```

```
npx create-next-app@latest brio-ia --typescript --tailwind --app
```

```
// 2. Estrutura de componentes
```

```
/app
```

```
  /auth
```

```
    /login
```

```
    /register
```

```
  /dashboard
```

```
    /chat
```

```
    /mpmp
```

```
    /content
```

```
  /api
```

```
    /auth
```

```
    /ai
```

```
    /content
```

```
// 3. Componente de Chat principal
```

```
// app/dashboard/chat/ChatInterface.tsx
```

```
'use client';
```

```
import { useState } from 'react';
```

```
import { Card } from '@components/ui/card';
```

```
import { Button } from '@components/ui/button';
```

```
import { Textarea } from '@components/ui/textarea';
```

```
export function ChatInterface() {
```

```
  const [prompt, setPrompt] = useState('');
```

```
  const [loading, setLoading] = useState(false);
```

```
  const [contents, setContents] = useState([]);
```

```
  const generateContent = async () => {
```

```
    setLoading(true);
```

```
    const response = await fetch('/api/ai/generate', {
```

```
      method: 'POST',
```

```
      headers: { 'Content-Type': 'application/json' },
```

```
      body: JSON.stringify({ prompt }),
```

```
    });
```

```
    const data = await response.json();
```

```
    setContents(data.variations);
```

```
    setLoading(false);
```

```
  };
```

```
  return (
```

```

<Card className="max-w-4xl mx-auto p-6">
  <h2 className="text-2xl font-bold mb-4">
    Criar Conteúdo Magnético
  </h2>

  <Textarea
    placeholder="Descreva o conteúdo que deseja criar..."
    value={prompt}
    onChange={(e) => setPrompt(e.target.value)}
    className="min-h-[100px] mb-4"
  />

  <Button
    onClick={generateContent}
    disabled={loading || !prompt}
    className="w-full"
  >
    {loading ? 'Gerando...' : 'Gerar Conteúdo'}
  </Button>

  {contents.map((content, index) => (
    <ContentCard key={index} content={content} />
  ))}
</Card>
);
}

```

## Sprint 3-4: Inteligência (Semanas 3-4)

### Objetivos

- Implementar os 8 códigos magnéticos
- Sistema de MPMP funcional
- Geração de conteúdo otimizada
- Interface melhorada

### Implementação dos 8 Códigos



```
// src/prompts/magnetic-codes.js
```

```
const MAGNETIC_CODES = {
  concordar_contrastar: {
    nome: "Concordar e Contrastar",
    descricao: "Validar para depois apresentar nova perspectiva",
    quando_usar: "Para educar sem confrontar",
    estrutura: "Sim, [crença comum] é importante, E TAMBÉM [nova perspectiva]",
    prompt_template: `
      Crie um conteúdo usando Concordar e Contrastar.
      Primeiro valide: {crenca_comum}
      Depois apresente: {nova_perspectiva}
      Tom: educativo e respeitoso
    `,
    exemplos: [
      "Sim, medicamentos são essenciais, E TAMBÉM...",
      "Concordo que dieta é importante, E TAMBÉM..."
    ]
  },

  atirar_pedras: {
    nome: "Atirar Pedras contra Inimigos",
    descricao: "Combater mitos e práticas ultrapassadas",
    quando_usar: "Para se posicionar contra o status quo",
    estrutura: "Enquanto [prática ruim] ainda é comum, a ciência mostra que [abordagem
    // ... continua para todos os 8 códigos
  }
};
```

```
// Seletor inteligente de código
```

```
class CodeSelector {
  selectBestCode(objective, context) {
    const codeScores = {};

    // Lógica de pontuação baseada em objetivo
    switch(objective) {
      case 'captar':
        codeScores['ruptura_cognitiva'] = 10;
        codeScores['elefante_sala'] = 8;
        break;
      case 'conectar':
        codeScores['historia_pessoal'] = 10;
        codeScores['confirmacao_suspeitas'] = 8;
        break;
      case 'convencer':
        codeScores['solucao_unica'] = 10;
        codeScores['vitoria_transformadora'] = 9;
    }
  }
}
```

```

        break;
    case 'converter':
        codeScores['vitoria_transformadora'] = 10;
        codeScores['solucao_unica'] = 9;
        break;
}

// Ajusta baseado no contexto do usuário
if (context.mppm?.expressao?.tom_de_voz?.energia === 'calma') {
    codeScores['concordar_contrastar'] += 3;
    codeScores['atirar_pedras'] -= 3;
}

return Object.entries(codeScores)
    .sort(([a], [b]) => b - a)
    .slice(0, 3)
    .map(([code]) => code);
}
}

```

## Sistema de MPMP Guiado



```
// app/dashboard/mpmp/MPMPWizard.tsx
```

```
const MPMP_STEPS = [  
  {  
    id: 'essencia',  
    title: 'Sua Essência',  
    questions: [  
      {  
        id: 'proposito',  
        question: 'Por que você escolheu a medicina/saúde?',  
        hint: 'Pense no momento que decidiu. O que te moveu?',  
        type: 'textarea'  
      },  
      {  
        id: 'valores',  
        question: 'Quais seus 3 valores inegociáveis?',  
        hint: 'O que você jamais comprometeria?',  
        type: 'multi-select',  
        options: [  
          'Ética', 'Inovação', 'Compaixão', 'Excelência',  
          'Transparência', 'Humanização', 'Ciência', 'Resultado'  
        ]  
      },  
      {  
        id: 'historia',  
        question: 'Qual história define quem você é como profissional?',  
        hint: 'Um momento transformador na sua jornada',  
        type: 'textarea'  
      }  
    ]  
  },  
  {  
    id: 'execucao',  
    title: 'Sua Execução',  
    questions: [  
      {  
        id: 'especialidade',  
        question: 'Qual sua especialidade principal?',  
        type: 'select',  
        options: ['Cardiologia', 'Endocrinologia', 'Psiquiatria', ...]  
      },  
      {  
        id: 'publico_idade',  
        question: 'Qual a faixa etária do seu público ideal?',  
        type: 'range',  
        min: 18,  
        max: 80  
      }  
    ]  
  }  
]
```



```

    },
    {
      id: 'dores_publico',
      question: 'Quais as 3 principais dores do seu público?',
      hint: 'O que tira o sono deles?',
      type: 'multi-input'
    }
  ]
},
{
  id: 'expressao',
  title: 'Sua Expressão',
  questions: [
    {
      id: 'tom_voz',
      question: 'Como você quer soar?',
      type: 'slider-matrix',
      dimensions: [
        { id: 'formalidade', min: 'Casual', max: 'Formal' },
        { id: 'energia', min: 'Calmo', max: 'Energético' },
        { id: 'proximidade', min: 'Profissional', max: 'Amigável' }
      ]
    }
  ]
}
];

```

*// Componente do Wizard*

```

export function MPMPWizard() {
  const [currentStep, setCurrentStep] = useState(0);
  const [answers, setAnswers] = useState({});

  const handleAnswer = (questionId, value) => {
    setAnswers(prev => ({
      ...prev,
      [currentStep]: {
        ...prev[currentStep],
        [questionId]: value
      }
    }));
  };

  const generateMPMP = async () => {
    const response = await fetch('/api/mpmp/generate', {
      method: 'POST',
      body: JSON.stringify(answers)
    });
  };
}

```

```
const mpmp = await response.json();  
// Salva e redireciona  
};  
}
```

## **Sprint 5-6: Experiência (Semanas 5-6)**

### **Objetivos**

- Interface polida e intuitiva
- Sistema de templates
- Onboarding otimizado
- Preparação para beta

### **Templates Inteligentes**



```
// src/data/templates.js
```

```
const CONTENT_TEMPLATES = [  
  {  
    id: 'desmistificador',  
    nome: 'Desmistificador de Mitos',  
    descricao: 'Derrube um mito comum da sua área',  
    codigo: 'atirar_pedras',  
    estrutura: 'MUS',  
    template: `  
❌ MITO: {{mito_comum}}
```

```
A verdade? {{verdade_cientifica}}
```

```
{{explicacao_detalhada}}
```

```
0 que isso significa para você:
```

```
✅ {{beneficio_1}}
```

```
✅ {{beneficio_2}}
```

```
✅ {{beneficio_3}}
```

```
{{chamada_para_acao}}
```

```
#{{especialidade}} #MitosDaSaude #{{hashtag_personalizada}}
```

```
`  
,  
  variaveis: [  
    {  
      id: 'mito_comum',  
      label: 'Qual mito você quer derrubar?',  
      placeholder: 'Ex: Ovo aumenta colesterol'  
    },  
    {  
      id: 'verdade_cientifica',  
      label: 'Qual a verdade científica?',  
      placeholder: 'Ex: Estudos mostram que...'  
    }  
  ]  
},
```

```
{  
  id: 'caso_sucesso',  
  nome: 'História de Transformação',  
  descricao: 'Compartilhe um caso de sucesso inspirador',  
  codigo: 'vitoria_transformadora',  
  estrutura: 'ENU',  
  template: `  
{{emoji_impacto}} TRANSFORMAÇÃO REAL
```

{{nome\_paciente}} chegou ao consultório {{situacao\_inicial}}.

{{jornada\_transformacao}}

Hoje, {{meses\_depois}} depois:

→ {{resultado\_1}}

→ {{resultado\_2}}

→ {{resultado\_3}}

{{licao\_aprendida}}

Você também pode transformar sua saúde.

{{chamada\_para\_acao}}

#{{especialidade}} #TransformacaoReal #Saude

`

}

];

*// Sistema de preenchimento inteligente*

class TemplateEngine {

fillTemplate(template, variables, userContext) {

let filled = template.template;

*// Preenche variáveis fornecidas*

Object.entries(variables).forEach(([key, value]) => {

filled = filled.replace(new RegExp(`{{\${key}}}`, 'g'), value);

});

*// Auto-preenche com contexto do usuário*

filled = filled.replace(/{{especialidade}}/g, userContext.especialidade);

filled = filled.replace(/{{chamada\_para\_acao}}/g,

this.generateCTA(template.codigo, userContext));

return filled;

}

generateCTA(codigo, context) {

*// Gera CTA personalizado baseado no código e contexto*

const ctas = {

'atirar\_pedras': 'Quer saber mais verdades? Siga para conteúdo baseado em ciência

'vitoria\_transformadora': 'Agende sua consulta e comece sua transformação hoje.'

*// ... mais CTAs*

};

return ctas[codigo] || 'Entre em contato para saber mais.';

}

}

**Dashboard Analítico**



```
// app/dashboard/analytics/ContentAnalytics.tsx
```

```
export function ContentAnalytics() {
  const [metrics, setMetrics] = useState(null);

  useEffect(() => {
    fetchMetrics();
  }, []);

  return (
    <div className="grid grid-cols-1 md:grid-cols-3 gap-6">
      <MetricCard
        title="Conteúdos Criados"
        value={metrics?.total_contents}
        change="+12% este mês"
        icon={<FileText />}
      />

      <MetricCard
        title="Tempo Economizado"
        value={` ${metrics?.hours_saved}h`}
        change="Equivale a 3 dias de trabalho"
        icon={<Clock />}
      />

      <MetricCard
        title="Código Mais Eficaz"
        value={metrics?.best_code}
        change={` ${metrics?.best_code_performance}% mais engajamento`}
        icon={<TrendingUp />}
      />

      <div className="col-span-full">
        <Card>
          <CardHeader>
            <CardTitle>Performance dos 8 Códigos</CardTitle>
          </CardHeader>
          <CardContent>
            <CodesPerformanceChart data={metrics?.codes_performance} />
          </CardContent>
        </Card>
      </div>

      <div className="col-span-full">
        <Card>
          <CardHeader>
            <CardTitle>Distribuição dos 4Cs</CardTitle>
          </CardHeader>
          <CardContent>
            <CodesPerformanceChart data={metrics?.codes_performance} />
          </CardContent>
        </Card>
      </div>
    </div>
  );
}
```



```
        <CardDescription>
            Equilíbrio entre Captar, Conectar, Convencer e Converter
        </CardDescription>
    </CardHeader>
    <CardContent>
        <FourCsDistribution data={metrics?.four_cs} />
    </CardContent>
</Card>
</div>
</div>
);
}
```

---

## Parte 3: Configurações Técnicas Detalhadas

### 3.1 Otimização de Custos de IA



```

// src/services/ai-optimizer.js
class AIOptimizer {
  constructor() {
    this.cache = new Redis();
    this.costPerToken = 0.00003; // Claude-3 pricing
  }

  async generateWithCache(prompt, context) {
    // Gera hash único para cache
    const cacheKey = this.generateCacheKey(prompt, context);

    // Verifica cache
    const cached = await this.cache.get(cacheKey);
    if (cached) {
      return { content: cached, fromCache: true, cost: 0 };
    }

    // Otimiza prompt para reduzir tokens
    const optimizedPrompt = this.optimizePrompt(prompt);

    // Estima custo antes de chamar
    const estimatedCost = this.estimateCost(optimizedPrompt);
    if (estimatedCost > context.user.maxCostPerRequest) {
      throw new Error('Custo estimado excede limite');
    }

    // Gera conteúdo
    const result = await this.ai.generate(optimizedPrompt);

    // Salva no cache com TTL baseado no tipo
    const ttl = this.getTTL(context.contentType);
    await this.cache.set(cacheKey, result, ttl);

    return {
      content: result,
      fromCache: false,
      cost: estimatedCost,
      tokensUsed: this.countTokens(optimizedPrompt + result)
    };
  }

  optimizePrompt(prompt) {
    // Remove redundâncias
    // Comprime instruções
    // Usa referências em vez de repetir contexto
    return optimizedPrompt;
  }
}

```

```
}

getTTL(contentType) {
  const ttls = {
    'post': 7 * 24 * 60 * 60, // 7 dias
    'template': 30 * 24 * 60 * 60, // 30 dias
    'mpmp': 60 * 60, // 1 hora (pode mudar)
  };
  return ttls[contentType] || 24 * 60 * 60;
}
}
```

## 3.2 Sistema de Filas para Processamento



```

// src/services/queue.service.js
const Queue = require('bull');
const contentQueue = new Queue('content-generation');

// Processador de filas
contentQueue.process(async (job) => {
  const { userId, prompt, context } = job.data;

  try {
    // Gera conteúdo
    const content = await aiService.generateContent(prompt, context);

    // Salva no banco
    await db.content.create({
      userId,
      content,
      metadata: job.data
    });

    // Notifica usuário
    await notificationService.send(userId, {
      type: 'content-ready',
      contentId: content.id
    });

    return { success: true, contentId: content.id };
  } catch (error) {
    // Retentar com backoff
    throw error;
  }
});

// API endpoint para adicionar à fila
app.post('/api/content/generate', async (req, res) => {
  const job = await contentQueue.add(req.body, {
    attempts: 3,
    backoff: {
      type: 'exponential',
      delay: 2000
    }
  });

  res.json({
    jobId: job.id,
    status: 'processing',
    estimatedTime: '30 seconds'
  });
});

```

```
});  
});
```

### 3.3 Monitoramento e Observabilidade

javascript

```
// src/monitoring/metrics.js
const prometheus = require('prom-client');

// Métricas customizadas
const contentGenerationDuration = new prometheus.Histogram({
  name: 'content_generation_duration_seconds',
  help: 'Duração da geração de conteúdo',
  labelNames: ['content_type', 'code', 'model']
});

const aiCostCounter = new prometheus.Counter({
  name: 'ai_cost_total_usd',
  help: 'Custo total com IA em USD',
  labelNames: ['user_id', 'model']
});

const activeUsersGauge = new prometheus.Gauge({
  name: 'active_users_total',
  help: 'Número de usuários ativos'
});

// Middleware de monitoramento
const monitoringMiddleware = (req, res, next) => {
  const start = Date.now();

  res.on('finish', () => {
    const duration = Date.now() - start;

    // Registra métricas HTTP
    httpDuration.observe(
      { method: req.method, route: req.route?.path, status: res.statusCode },
      duration / 1000
    );
  });

  next();
};

// Dashboard de métricas
app.get('/metrics', async (req, res) => {
  res.set('Content-Type', prometheus.register.contentType);
  res.end(await prometheus.register.metrics());
});
```

---



## **Parte 4: Checklists de Implementação**

### **Checklist Pre-Launch**

#### **Segurança**

- ☐ Autenticação JWT implementada
- ☐ Rate limiting configurado
- ☐ CORS configurado corretamente
- ☐ Variáveis de ambiente seguras
- ☐ SQL injection prevenido (Prisma)
- ☐ XSS prevenido (React)
- ☐ Criptografia de dados sensíveis
- ☐ Backup automático configurado

#### **Performance**

- ☐ Cache Redis funcionando
- ☐ Compressão gzip ativada
- ☐ Imagens otimizadas
- ☐ Lazy loading implementado
- ☐ Bundle size < 200KB
- ☐ Time to First Byte < 200ms
- ☐ Lighthouse score > 90

#### **Monitoramento**

- ☐ Sentry configurado para erros
- ☐ Analytics configurado
- ☐ Logs estruturados
- ☐ Alertas configurados
- ☐ Dashboard de métricas
- ☐ Uptime monitoring

#### **Compliance**

- ☐ Termos de uso
- ☐ Política de privacidade
- ☐ LGPD compliance
- ☐ Aviso de uso de IA
- ☐ Disclaimer médico
- ☐ Processo de exclusão de dados

### **Checklist Pós-Launch (Primeira Semana)**

## Dia 1

- ☐ Monitorar erros em tempo real
- ☐ Verificar performance da IA
- ☐ Responder feedback urgente
- ☐ Verificar custos de API

## Dia 2-3

- ☐ Analisar primeiros usos
- ☐ Identificar pontos de fricção
- ☐ Ajustar prompts baseado em outputs
- ☐ Otimizar queries lentas

## Dia 4-7

- ☐ Compilar feedback dos usuários
- ☐ Priorizar melhorias
- ☐ Planejar primeira atualização
- ☐ Celebrar launch! 🎉

---

## Parte 5: Recursos e Referências

### APIs e Documentação

- [Anthropic API \(Claude\)](#).
- [Prisma ORM](#)
- [Next.js 14](#)
- [Shadcn/ui](#)
- [Stripe API](#)

### Ferramentas de Desenvolvimento

- [Postman](#) - Testar APIs
- [Prisma Studio](#) - Visualizar banco
- [Ray](#) - Gerar imagens de código
- [Excalidraw](#) - Diagramas

### Comunidades e Suporte

- Discord da Anthropic
- r/nextjs no Reddit
- Stack Overflow

## 💡 **Notas Finais**

Este blueprint é um documento vivo. À medida que o Agente BRIO.IA evolui, este documento deve ser atualizado para refletir novas aprendizagens, melhores práticas descobertas e feedback dos usuários.

Lembre-se: O objetivo não é criar o sistema perfeito desde o início, mas sim um sistema que aprende e melhora continuamente, sempre mantendo o foco em gerar valor real para os profissionais de saúde.

**Próxima revisão:** 30 dias após o lançamento **Versão:** 1.0 **Última atualização:** [Data atual]