

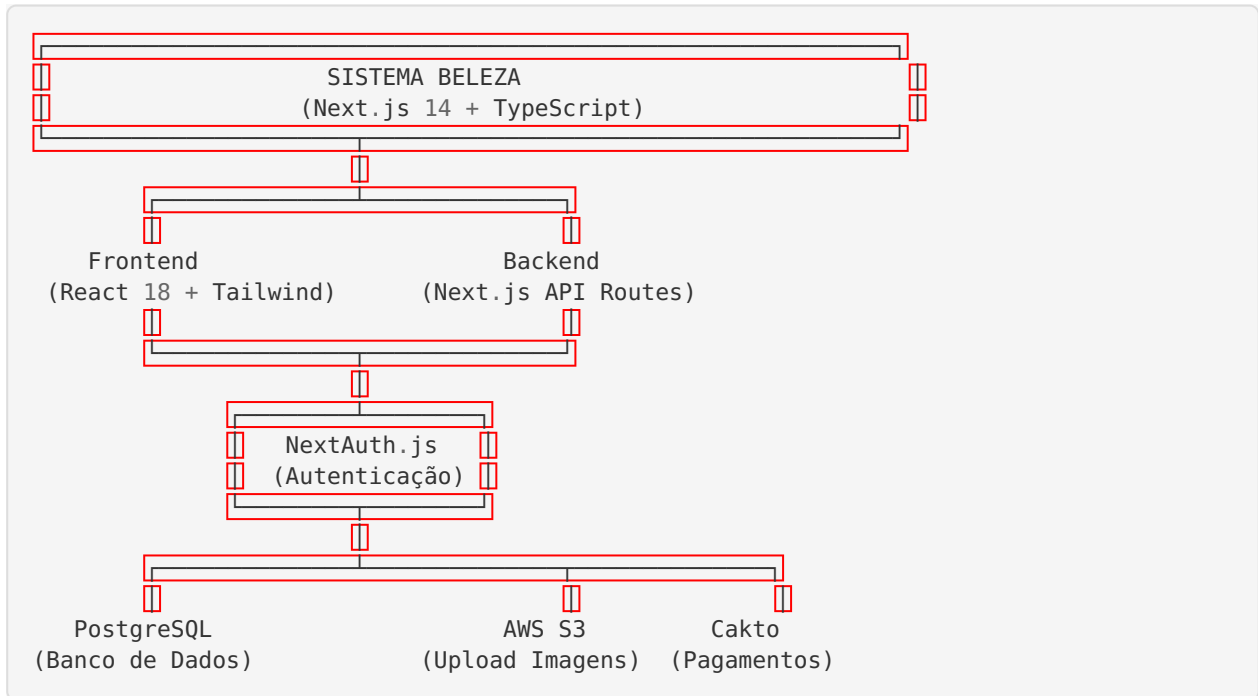


Análise Completa: Sistema Beleza



ESTRUTURA ATUAL DO SISTEMA

Arquitetura Multi-Tenant



Estrutura de Pastas

sistema_salao_beleza/	
nextjs_space/	# Aplicação Next.js
app/	# App Router do Next.js
page.tsx	# Landing page
auth/	# Login e Signup
dashboard/	# Painel principal
page.tsx	# Dashboard principal
agenda/	# Gestão de agendamentos
clientes/	# Gestão de clientes
profissionais/	# Gestão de profissionais
servicos/	# Gestão de serviços
produtos/	# Gestão de produtos
vendas/	# PDV e vendas
financeiro/	# Gestão financeira
relatorios/	# Relatórios e gráficos
configuracoes/	# Configurações do salão
agendamento/[slug]/	# Agendamento público
admin /	# Painel administrativo
api/	# APIs do sistema
components/	# Componentes React
ui/	# Componentes Shadcn/ui
dashboard-layout.tsx	# Layout principal
lib/	# Utilitários e configs
auth.ts	# Configuração NextAuth
db.ts	# Cliente Prisma
s3.ts	# Upload AWS S3
plan -limits.ts	# Limites por plano
utils.ts	# Funções auxiliares
prisma/	# Configuração do banco
schema .prisma	# Schema do banco
migrations/	# Migrations
public /	# Arquivos estáticos
logo-sistema-beleza.png	
favicon.svg	
package.json	# Dependências
next .config.js	# Config Next.js
tailwind.config.ts	# Config Tailwind
tsconfig.json	# Config TypeScript
README.md	# Documentação principal
GITHUB_SETUP.md	# Guia GitHub
ANALISE_SISTEMA.md	# Este arquivo
.gitignore	# Arquivos ignorados

BANCO DE DADOS - CAPACIDADE

Configuração PostgreSQL

Tipo: PostgreSQL Hosted Database
Host: db-42302409.db002.hosteddb.reai.io
Port: 5432
Conexões Simultâneas: 100-200 (padrão)
Timeout: 15 segundos

Capacidade por Métricas

1. Usuários Simultâneos

CONFIGURAÇÃO ATUAL

- 100-200 usuários simultâneos
- Cada usuário = 1 conexão ativa ao banco
- Connection pooling automático do Prisma

ESCALABILIDADE


Com PgBouncer: 500-1.000 usuários simultâneos
Com Redis Cache: 1.000-2.000 usuários simultâneos
Com Load Balancer: 5.000+ usuários simultâneos

2. Salões (Tenants)

Capacidade Atual: 5.000+ salões
Capacidade Máxima: Ilimitado (depende apenas do storage)

Exemplo de Crescimento:

Salões	Clientes	Agendamentos	Armazenamento
100	10.000	100.000	~500 MB
500	50.000	500.000	~2.5 GB
1.000	100.000	1.000.000	~5 GB
5.000	500.000	5.000.000	~25 GB
10.000	1.000.000	10.000.000	~50 GB

 PostgreSQL suporta facilmente bancos de 100+ GB

3. Dados por Salão

Limites por Salão (baseados no plano):

Recurso	BÁSICO	INTERMEDIÁRIO	COMPLETO
Profissionais	2	5	∞
Serviços	10	30	∞
Agendamentos/mês	100	500	∞
Clientes	∞	∞	∞
Produtos	∞	∞	∞
Vendas	∞	∞	∞
Controle Financeiro	Básico	Completo	Completo
Relatórios	Simple	Avançados	Avançados
Agendamento Público	X	✓	✓

Capacidade Real (sem limites de plano):

- Clientes: 10.000+ por salão
- Agendamentos: 50.000+ por salão
- Produtos: 1.000+ por salão
- Vendas: 20.000+ por salão

4. Performance de Queries

Queries Otimizadas com **I**ndices:

Operação	Tempo Médio	Registros
Login/Autenticação	50-100ms	-
Listar Agendamentos (dia)	20-50ms	10-50
Criar Agendamento	30-80ms	1
Buscar Cliente	10-30ms	1
Listar Produtos	30-70ms	100-500
Registrar Venda	50-150ms	1 + itens
Dashboard (KPIs)	100-300ms	agregações
Relatório Mensal	200-500ms	1000-5000

💡

Todas **as** queries usam **I**ndices otimizados

💡

Tempo pode variar com a carga do servidor

5. Armazenamento AWS S3

UPLOADS DE IMAGENS

Bucket: abacusai-apps-c20175eafe99c22609c6d07e-us-west-2
 Região: us-west-2
 Capacidade: ILIMITADA
 Custo: ~\$0.023/GB/mês

Tipos Aceitos:

- Logos dos salões
- Fotos dos salões (2 fotos)
- Fotos de clientes
- Fotos de profissionais
- Fotos de produtos

Estimativa de Uso:

- 1 salão = ~3 imagens (logo + 2 fotos) = 1-3 MB
- 100 clientes = ~50 fotos = 10-20 MB
- 10 profissionais = ~10 fotos = 2-5 MB
- 100 produtos = ~50 fotos = 10-20 MB

Total por salão médio: 30-50 MB
 1.000 salões: ~40 GB



ESCALABILIDADE

Plano de Crescimento

FASE 1: 0-100 Salões (Configuração Atual)

- ☒ PostgreSQL padrão (100-200 conexões)
- ☒ AWS S3 para uploads
- ☒ Next.js em servidor ☒ único
- ☐ Capacidade: 200 usuários simultâneos

FASE 2: 100-500 Salões

- ☐ PgBouncer (connection pooling)
- ☐ Redis para cache de queries
- ☒ CDN para assets estáticos (Cloudflare)
- ☐ Capacidade: 500-1.000 usuários simultâneos

FASE 3: 500-2.000 Salões

- ☐ PostgreSQL upgrade (mais RAM/CPU)
- ☐ Load Balancer (distribuir carga)
- ☐ Read Replicas (queries de leitura)
- ☐ Capacidade: 2.000-5.000 usuários simultâneos

FASE 4: 2.000+ Salões (Enterprise)

- ☐ Kubernetes (auto-scaling)
- ☐ PostgreSQL Cluster (alta disponibilidade)
- ☐ ElastiCache Redis Cluster
- ☐ Microservices (separar funcionalidades)
- ☐ Capacidade: 10.000+ usuários simultâneos

Custos Estimados (Infraestrutura)

Fase	Salões	Usuários/mês	Custo/mês
Fase 1 (Atual)	0-100	1.000	\$50-100
Fase 2	100-500	5.000	\$200-400
Fase 3	500-2.000	20.000	\$800-1.500
Fase 4	2.000+	50.000+	\$2.000-5.000

💡 Custos incluem: Servidor, Banco, Storage, CDN

💡 Não incluem: Marketing, Suporte, Desenvolvimento



MONITORAMENTO RECOMENDADO

Métricas Essenciais

Performance:

- Tempo de resposta das APIs (< 500ms)
- Tempo de carregamento de páginas (< 2s)
- Taxa de erro (< 1%)

Banco de Dados:

- Conexões ativas (< 80% do máximo)
- Queries lentas (> 1s)
- Tamanho do banco (crescimento)

Negócio:

- Novos salões cadastrados/dia
- Taxa de conversão (trial → pago)
- Churn rate (cancelamentos)
- Agendamentos/dia

Ferramentas Recomendadas

Monitoramento:

- └ Vercel Analytics (performance frontend)
- └ Prisma Studio (visualizar banco)
- └ AWS CloudWatch (logs S3)
- └ Google Analytics (comportamento usuário)

Alertas:

- └ **Uptime** Robot (disponibilidade)
- └ Sentry (erros JavaScript)
- └ Email/SMS quando:
 - └ Site fora do ar > 5 min
 - └ Erro crítico no banco
 - └ Uso > 90% capacidade






RESUMO EXECUTIVO

Capacidade Atual (Sem Upgrades)

- ✓ 100-200 usuários simultâneos
- ✓ 5.000+ salões cadastrados
- ✓ 50.000+ clientes totais
- ✓ 100.000+ agendamentos/mês
- ✓ Storage ilimitado (AWS S3)
- ✓ Multi-tenancy seguro (zero vazamento de dados)

Quando Fazer Upgrade

SINAIS DE ALERTA:

1. Conexões do Banco > 80%
 Solução: Implementar PgBouncer
2. Queries lentas (> 1s)
 Solução: Adicionar Redis cache
3. Site lento em horários de pico
 Solução: Load Balancer + CDN
4. Banco de dados > 50 GB
 Solução: Upgrade plan PostgreSQL
5. Uploads lentos
 Solução: CloudFront CDN para S3

Recomendações Imediatas

1. ✓ Implementar monitoramento básico
2. ✓ Configurar backups automáticos diários
3. ✓ Documentar procedimentos de recuperação
4. ✓ Testar carga (stress testing)
5. ✓ Configurar alertas de capacidade

Suporte Técnico

Para dúvidas sobre capacidade e escalabilidade:

Email: sistemabeleza.contato@gmail.com

Última Atualização: Novembro 2025

Versão: 1.0.0