

ROADMAP DE PRODUTO

Sistema de Agendamento para Fotografia Imobiliária

Versão: 1.0

Data: Janeiro 2026

Empresa: Nome_da_Empresa

Produto: Plataforma de Agendamento Online

VISÃO DO PRODUTO

Objetivo

Criar uma plataforma web que permita clientes agendarem sessões de fotografia imobiliária de forma 100% automatizada, sem necessidade de contato prévio, otimizando a alocação de 4 fotógrafos em Curitiba.

Problema que Resolve

- **Clientes:** Dificuldade em agendar serviços (dependência de atendimento, horário comercial)
- **Empresa:** Sobrecarga operacional (atendimento manual, conflitos de agenda, erros de alocação)
- **Fotógrafos:** Falta de visibilidade da agenda, deslocamentos ineficientes

Proposta de Valor

- Agendamento 24/7 sem intermediários
 - Confirmação instantânea
 - Alocação inteligente de fotógrafos
 - Transparência total (horários, serviços, preços)
 - Redução de 80% do tempo de atendimento
-

FASE 0: SITUAÇÃO ATUAL

Status: Protótipo Funcional Completo (95%)

O que já temos:

- Interface web completa (6 passos de agendamento)
- Design mobile-first responsivo
- Validação de endereço (simulada)
- Seleção múltipla de serviços (5 opções)
- Calendário formato padrão (7 dias/semana)
- Cálculo automático de duração e slots
- Bloqueio inteligente de horários consecutivos
- Fluxo de confirmação com protocolo
- Mensagens de erro e validação

Gaps para MVP:

- Integração Google Maps API (endereços reais)
 - Banco de dados real (PostgreSQL)
 - APIs backend funcionais
 - Envio de emails/WhatsApp
 - Sincronização com Tadabase
 - Deploy em produção
-

FASE 1: MVP (PRODUTO MÍNIMO VIÁVEL)

Timeline: 3-4 semanas

Objetivo: Sistema funcionando em produção com funcionalidades essenciais

Funcionalidades Core

1.1 Agendamento pelo Cliente

Prioridade: CRÍTICA

User Story:

Como cliente, quero agendar uma sessão fotográfica online sem precisar falar com ninguém, para economizar tempo e ter confirmação imediata.

Critérios de Aceite:

- Cliente acessa URL e completa agendamento em < 3 minutos
- Sistema valida endereço via Google Maps API
- Sistema mostra apenas horários realmente disponíveis
- Sistema bloqueia horário por 10min durante processo
- Cliente recebe email de confirmação em < 1 minuto
- Sistema gera protocolo único

Fluxo:

1. Inserir endereço → Validar cobertura
2. Selecionar serviço(s) → Calcular duração
3. Escolher data → Filtrar dias disponíveis
4. Escolher horário → Ver slots compatíveis com duração
5. Preencher dados → Validar campos
6. Confirmar → Receber protocolo

Telas:

- Passo 1: Endereço + Complemento
- Passo 2: Serviços (checkbox múltiplo)
- Passo 3: Calendário (7 dias/semana, domingos desabilitados)
- Passo 4: Horários (com início e fim)
- Passo 5: Dados do cliente (nome, email, telefone, obs)
- Passo 6: Confirmação (resumo completo)
- Passo 7: Sucesso (protocolo + instruções)

1.2 Validação de Área de Cobertura

Prioridade: CRÍTICA

Regras:

- **Whitelist:** Lista de 15-30 bairros de Curitiba (aprovação instantânea)
- **Margem:** +3km além do último bairro (aprovação com validação)
- **Blacklist:** Municípios da RMC (rejeição com link WhatsApp)

Integração:

- Google Maps Geocoding API
- Google Distance Matrix API

Mensagens:

- "Atendemos sua região! Continue o agendamento"
 - "Verificando disponibilidade na sua região..."
 - "Esta região ainda não é atendida. Entre em contato: (41) 9999-9999"
-

1.3 Gestão Inteligente de Horários**Prioridade:** CRÍTICA**Regras:**

- Horários de 30 em 30 minutos
- Segunda a Sexta: 08:00 - 17:30
- Sábado: 08:00 - 13:00
- Domingo: Não atende
- Não permite agendamento no mesmo dia (D+1 mínimo)

Cálculo de Slots:

- Fotos (40min) = 2 slots de 30min
- Vídeo Paisagem (50min) = 2 slots
- Vídeo Retrato (50min) = 2 slots
- Drone Fotos (25min) = 1 slot
- Drone Fotos+Vídeo (40min) = 2 slots
- Múltiplos serviços = soma das durações

Bloqueio Automático:

- Se sessão dura 80min (3 slots), sistema bloqueia:
 - Slot escolhido

- Próximo slot
- Slot seguinte
- Exemplo: Agendou 09:00 com 80min → bloqueia 09:00, 09:30, 10:00

Disponibilidade:

- Sistema verifica distância do fotógrafo do agendamento anterior
- Se > 30min de deslocamento → horário indisponível
- Se todos fotógrafos ocupados → horário indisponível

1.4 Alocação de Fotógrafos

Prioridade: CRÍTICA

Equipe:

- **Augusto:** Foto, Vídeo
- **Renato:** Foto
- **Rafael:** Foto, Vídeo, Drone
- **Rodrigo:** Foto

Lógica de Alocação:

SE serviço requer DRONE:
→ Apenas Rafael disponível

SE serviço requer VÍDEO (paisagem ou retrato):
→ Augusto ou Rafael

SE serviço é APENAS FOTO:
→ Qualquer um dos 4

Priorizar fotógrafo:
1. Mais próximo do local (menor deslocamento)
2. Com menos agendamentos no dia (balanceamento)
3. Ordem alfabética (desempate)

1.5 Notificações Automáticas

Prioridade: ALTA

Email de Confirmação:

- Enviado em < 1 minuto após agendamento
- Contém:
 - Protocolo
 - Data, horário, duração
 - Endereço completo
 - Serviços contratados
 - Nome do fotógrafo
 - Link de cancelamento
 - Instruções de preparação

Notificação ao Fotógrafo:

- Email imediato com detalhes
- Informações do cliente
- Localização no mapa

Lembretes (Fase 2):

- 24h antes: WhatsApp para cliente e fotógrafo
- 2h antes: WhatsApp de confirmação final

1.6 Sincronização Bidirecional Tadabase

Prioridade: CRÍTICA

Fluxo Sistema → Tadabase:

```
Cliente agenda no site  
↓  
Sistema salva no banco local  
↓  
Sistema envia webhook para Tadabase/Make.com  
↓  
Tadabase registra agendamento  
↓  
Retorna confirmação
```

Fluxo Tadabase → Sistema:

```
Atendente agenda manualmente no Tadabase  
↓  
Tadabase envia webhook para sistema  
↓  
Sistema valida disponibilidade  
↓  
SE disponível: cria agendamento + bloqueia horário  
SE ocupado: retorna erro 409 (conflito)
```

Tratamento de Conflitos:

- Rejeita agendamento duplicado
- Notifica admin sobre tentativa
- Log completo no banco

Tecnologias do MVP

Frontend

- **Framework:** Next.js 14 (App Router)
- **Linguagem:** TypeScript
- **Estilo:** Tailwind CSS
- **Componentes:** Shaden/ui
- **Validação:** React Hook Form + Zod
- **Ícones:** Lucide React

Backend

- **Runtime:** Next.js API Routes (Serverless)
- **Banco de Dados:** Vercel Postgres (256MB free tier)
- **Cache:** Vercel KV (Redis)
- **ORM:** Prisma

Integrações

- **Maps:** Google Maps JavaScript API, Places API, Distance Matrix API
- **Email:** Resend (100 emails/dia grátis)
- **Webhook:** Tadabase ou Make.com
- **Analytics:** Vercel Analytics

Infraestrutura

- **Hospedagem:** Vercel (Free tier)
 - **Domínio:** Cloudflare (opcional)
 - **SSL:** Vercel (automático)
 - **Backup:** Postgres diário automático
-

Estrutura do Banco de Dados (MVP)

sql

```
-- Fotógrafos
photographers
└── id (UUID)
└── name (VARCHAR)
└── email (VARCHAR)
└── phone (VARCHAR)
└── services (JSON) -- ["photo", "video", "drone"]
└── active (BOOLEAN)
└── created_at (TIMESTAMP)

-- Áreas de cobertura
coverage_areas
└── id (UUID)
└── type (VARCHAR) -- 'neighborhood', 'exclusion'
└── value (VARCHAR) -- nome do bairro
└── active (BOOLEAN)
└── created_at (TIMESTAMP)

-- Agendamentos
bookings
└── id (UUID)
└── protocol (VARCHAR) -- AG20260122001
└── photographer_id (UUID FK)
└── source (VARCHAR) -- 'web', 'tadabase'
└── client_name (VARCHAR)
└── client_email (VARCHAR)
└── client_phone (VARCHAR)
└── service_type (JSON) -- ["photo", "video_landscape"]
└── duration_minutes (INT)
└── address_full (TEXT)
└── address_lat (DECIMAL)
└── address_lng (DECIMAL)
└── scheduled_date (DATE)
└── scheduled_start (TIME)
└── scheduled_end (TIME)
└── status (VARCHAR) -- 'confirmed', 'cancelled', 'completed'
└── cancellation_token (VARCHAR)
└── tadabase_synced (BOOLEAN)
└── tadabase_id (VARCHAR)
└── created_at (TIMESTAMP)
└── updated_at (TIMESTAMP)
```

```
-- Logs de webhook
webhook_logs
├── id (UUID)
├── source (VARCHAR) -- 'tadabase'
├── event_type (VARCHAR)
├── payload (JSON)
├── status (VARCHAR) -- 'success', 'failed'
└── error_message (TEXT)
└── created_at (TIMESTAMP)
```

APIs do MVP

Públicas (Cliente)

```
POST /api/address/validate
├── Input: { address: string }
└── Output: { valid: boolean, neighborhood: string, lat: number, lng: number }

GET /api/availability
├── Query: date, services[]
└── Output: { slots: [{ time, endTime, available }] }

POST /api/bookings
├── Input: { address, services, date, time, client: {...} }
└── Output: { protocol, booking_id }

DELETE /api/bookings/:token
├── Cancela agendamento via link único
└── Output: { cancelled: boolean }
```

Internas (Webhook)

```
POST /api/webhooks/tadabase
├── Recebe agendamentos criados no Tadabase
├── Valida assinatura HMAC
├── Verifica disponibilidade
└── Cria ou rejeita agendamento
```

```
POST /api/webhooks/send-to-tadabase
├── Envia agendamento para Tadabase
└── Retry automático em caso de falha
```

Métricas de Sucesso (MVP)

Técnicas:

- Tempo de resposta < 2s em 95% das requisições
- Uptime > 99%
- Zero conflitos de agendamento
- 100% de sincronização com Tadabase

Negócio:

- 80% dos agendamentos via web (vs telefone/WhatsApp)
- Taxa de conclusão do fluxo > 70%
- Taxa de cancelamento < 10%
- Tempo médio de agendamento < 3min

UX:

- Cliente consegue agendar sozinho sem dúvidas
- Zero reclamações sobre endereço não encontrado
- Emails chegam em < 1min

FASE 2: FUNCIONALIDADES AVANÇADAS

Timeline: +2-3 semanas após MVP

Objetivo: Melhorar autonomia e experiência

2.1 Painel do Fotógrafo

Prioridade: ALTA

Funcionalidades:

- Login individual (email + senha)
- Dashboard pessoal:
 - Agendamentos do dia/semana
 - Próximo agendamento (countdown)
 - Total de km rodados no dia
- Gestão de disponibilidade:
 - Bloquear dia completo (férias, folga)
 - Bloquear horários específicos (compromisso, manutenção)
 - Desbloquear horários
- Visualizar rotas no mapa
- Histórico de sessões

Telas:

- Login
- Dashboard
- Calendário pessoal
- Detalhes do agendamento
- Configurações

2.2 Cancelamento pelo Cliente

Prioridade: ALTA

Regras:

- **> 24h antes:** Cancelamento gratuito
- **12h - 24h:** Taxa de 50%
- **< 12h:** Taxa de 100%

- < 2h: Não permite cancelamento online (ligar)

Fluxo:

- Cliente clica link no email
- Sistema valida token único
- Mostra resumo do agendamento
- Exibe prazo e taxa aplicável
- Cliente confirma cancelamento
- Sistema:
 - Atualiza status no banco
 - Libera horário
 - Notifica fotógrafo
 - Envia email de confirmação
 - Atualiza Tadabase

2.3 Reagendamento pelo Cliente

Prioridade: MÉDIA

Regras:

- Mesmas regras de cancelamento aplicam
- Pode escolher nova data/horário
- Pode fazer upgrade de serviço
- Não pode fazer downgrade
- Mantém mesmo protocolo

Fluxo:

- Cliente clica "Reagendar" no email
- Sistema mostra horários disponíveis
- Cliente escolhe novo slot
- Sistema valida disponibilidade
- Confirma reagendamento

- Envia nova confirmação
-

2.4 WhatsApp Notifications

Prioridade: ALTA

Via: Twilio WhatsApp Business API

Mensagens:

[Confirmação Imediata]

- ✓ Agendamento confirmado!
- 📅 25/01/2026 às 09:00
- 📍 Rua XV, 1000 - Centro
- 🕒 Duração: 80min
- 🔢 Protocolo: AG20260122001

[Lembrete 24h antes]

- ⏰ Lembrete: Sessão amanhã!
- 📅 26/01 às 09:00
- 📍 Rua XV, 1000
- 👤 Fotógrafo: Rafael

[Lembrete 2h antes]

- ⌚ Sua sessão começa em 2 horas!
- Até logo! 📷

2.5 Sistema de Lembretes Automáticos

Prioridade: MÉDIA

Agendamento de Jobs:

- Cron job roda 3x ao dia
- Verifica agendamentos próximos
- Envia lembretes via WhatsApp/Email

Timeline:

- **D-1 (24h antes):** Lembrete ao cliente e fotógrafo
 - **D (2h antes):** Lembrete final
 - **D (após sessão):** Pedir feedback (Fase 3)
-

FASE 3: ADMINISTRAÇÃO E ANALYTICS

Timeline: +2-3 semanas após Fase 2

Objetivo: Gestão eficiente e tomada de decisão

3.1 Dashboard Administrativo

Prioridade: ALTA

KPIs Principais:

- Agendamentos hoje/semana/mês
- Taxa de ocupação por fotógrafo
- Regiões mais demandadas
- Serviços mais solicitados
- Horários de pico
- Taxa de cancelamento
- Receita estimada

Gráficos:

- Agendamentos por dia (últimos 30 dias)
 - Distribuição por serviço (pizza)
 - Heatmap de horários populares
 - Mapa de calor geográfico (Curitiba)
-

3.2 Gestão de Fotógrafos

Prioridade: MÉDIA

Funcionalidades:

- Adicionar novo fotógrafo
 - Editar dados (nome, telefone, serviços)
 - Desativar fotógrafo (sem excluir histórico)
 - Ver estatísticas individuais
 - Ajustar regiões de atuação
-

3.3 Gestão de Regiões

Prioridade: MÉDIA

Funcionalidades:

- Adicionar bairro à whitelist
 - Remover bairro
 - Ajustar margem de cobertura (km)
 - Adicionar município à blacklist
 - Visualizar mapa de cobertura
-

3.4 Relatórios e Exportação

Prioridade: BAIXA

Relatórios:

- Agendamentos por período (Excel/CSV)
- Relatório financeiro (estimativa)
- Relatório de deslocamento (km rodados)
- Análise de demanda por região

Automação:

- Relatório semanal por email
 - Alertas de baixa ocupação
 - Alertas de horários sempre cheios
-

FASE 4: OTIMIZAÇÕES E ESCALABILIDADE

Timeline: +1-2 semanas

Objetivo: Performance e preparação para crescimento

4.1 Cache Inteligente

- Cache de bairros (Vercel KV)
- Cache de horários disponíveis (5min)
- Cache de geocodificação (permanente)

4.2 Otimização de Queries

- Índices no banco de dados
- Query optimization
- Connection pooling

4.3 SEO e Performance

- Meta tags otimizadas
- Sitemap.xml
- Schema.org markup
- Lazy loading de imagens
- Code splitting

4.4 PWA (Progressive Web App)

- Instalável no smartphone
- Funciona offline (básico)
- Push notifications

FASE 5: EXPANSÃO (FUTURO)

5.1 Multi-Cidade

- Suporte a outras cidades

- Gestão de múltiplas equipes
- Configuração por região

5.2 Pagamento Online

- Integração Stripe/Mercado Pago
- Pagamento na confirmação
- Estorno automático em cancelamento

5.3 Sistema de Avaliações

- Cliente avalia fotógrafo
- Nota de 1-5 estrelas
- Comentários
- Badge de excelência

5.4 Integração Google Calendar

- Sincronização automática
- Fotógrafo vê agenda no Google
- Atualizações bidirecionais

5.5 App Mobile Nativo

- iOS e Android
- Notificações push nativas
- Experiência otimizada

RISCOS E MITIGAÇÕES

Risco	Probabilidade	Impacto	Mitigação
Google Maps API custo alto	Média	Alto	Cache agressivo + limite mensal
Vercel free tier insuficiente	Média	Médio	Monitorar uso + upgrade se necessário
Conflitos de agendamento	Alta	Crítico	Lock temporário + validações

Risco	Probabilidade	Impacto	Mitigação
Falha integração Tadabase	Média	Alto	Retry automático + fila
Cliente não recebe email	Baixa	Alto	Double-check + logs + retry
Sincronização web↔tadabase	Alta	Crítico	Webhook com assinatura + logs

DEPENDÊNCIAS EXTERNAS

Críticas (bloqueiam MVP):

- Google Cloud Platform (Maps API)
- Vercel (hospedagem)
- Resend ou similar (emails)
- Tadabase API ou Make.com

Importantes (podem ser substituídas):

- Twilio (WhatsApp) - pode ser manual inicialmente
- Analytics - pode usar Google Analytics grátis

CRITÉRIOS DE SUCESSO POR FASE

MVP Lançado com Sucesso SE:

- 100 agendamentos sem erro nos primeiros 30 dias
- Zero conflitos de horário
- 100% de sincronização Tadabase
- < 5% de reclamações sobre UX
- Tempo médio de agendamento < 3min

Fase 2 com Sucesso SE:

- 80% dos fotógrafos usam painel
- Taxa de cancelamento < 8%

- WhatsApp com 99% de entrega

Fase 3 com Sucesso SE:

- Admin usa dashboard semanalmente
 - Decisões baseadas em dados
 - Identificação de oportunidades (horários/regiões)
-

ANEXOS

A. Glossário

- **Slot:** Intervalo de 30 minutos
- **Lock:** Bloqueio temporário durante agendamento
- **Whitelist:** Bairros atendidos automaticamente
- **Blacklist:** Municípios não atendidos
- **Protocolo:** Código único (ex: AG20260122001)
- **Webhook:** Notificação automática entre sistemas

B. Links Úteis

- Documentação Google Maps API
 - Documentação Vercel
 - Documentação Tadabase
 - Repositório Git (quando criado)
-

Documento vivo - atualizar conforme evolução do produto

Última atualização: Janeiro 2026