

# Product Requirements Document (PRD)

## Vitrine de Alimentos via WhatsApp

---

### 1. Informações Gerais

- **Nome do Produto/Projeto:** Vitrine de Alimentos via WhatsApp
  - **Responsável (PO/PM):** [Definir responsável]
  - **Data:** [dd/mm/aaaa]
  - **Versão:** 1.0
- 

### 2. Visão Geral

#### Contexto

O sistema será uma plataforma para cadastro e gerenciamento de produtos alimentícios (ex: Lanches, Pizzas, Batatas Recheadas e Salgados em Geral), com possibilidade de expansão para outros tipos de alimentos futuramente.

Diferente de um e-commerce tradicional com vitrine e checkout, o foco será um agente inteligente no WhatsApp, que irá:

- Responder automaticamente os clientes sobre o cardápio disponível
- Informar preços e descrições dos produtos
- Interagir de forma natural, ajudando o cliente a consultar rapidamente o menu

O administrador continuará tendo acesso a um painel de cadastro e configuração, onde poderá:

- Inserir, editar e remover produtos
- Definir preços, descrições e disponibilidade
- Configurar parâmetros do agente (mensagem inicial, tom de voz, respostas automáticas)

#### Objetivo

Criar uma solução de atendimento automatizado via WhatsApp que permita aos estabelecimentos de alimentação disponibilizar seu cardápio de forma prática, sem depender de marketplaces como iFood. O sistema oferecerá controle total do cardápio via painel e atendimento rápido e direto via agente automatizado no WhatsApp.

#### Stakeholders

- Product Owner (PO)
  - Time de Desenvolvimento (Backend, Frontend, Integração com WhatsApp, IA/NLP)
  - Design (UX Conversacional)
  - Cliente Final (Estabelecimentos)
  - Consumidores (usuários via WhatsApp)
- 

### 3. Problema / Oportunidade

#### Dores/necessidades

- Atendimento manual no WhatsApp causa demora e perda de clientes
- Marketplaces têm taxas altas e pouco controle de experiência
- Sobrecarga de donos de negócio por falta de automação

#### Oportunidade

- Criar agente automatizado no WhatsApp
- Fornecer painel simples para gestão de cardápio
- Canal próprio de atendimento direto

#### Impacto se não resolvermos

- Perda de clientes por demora
  - Dependência de marketplaces
  - Limitação de crescimento
- 

### 4. Objetivos do Produto

- Criar agente no WhatsApp para responder sobre cardápio, preços e disponibilidade
  - Painel administrativo para cadastro e gestão de produtos
  - Configuração do agente (boas-vindas, tom de voz, respostas padrão)
  - Reduzir tempo de resposta, aumentar conversão
  - Base escalável para integrações futuras (pagamentos, delivery, fidelidade)
- 

### 5. Requisitos Funcionais

#### Cadastro e Gestão de Produtos

- **RF-01:** Cadastrar produtos (nome, descrição, preço, categoria, foto, disponibilidade)

- **RF-02:** Editar produtos existentes
- **RF-03:** Remover produtos
- **RF-04:** Organizar produtos por categorias
- **RF-05:** Marcar produtos como disponíveis/indisponíveis

### **Configuração do Agente**

- **RF-06:** Definir mensagem de boas-vindas
- **RF-07:** Configurar tom de voz
- **RF-08:** Configurar respostas padrão
- **RF-09:** Ativar/desativar funcionalidades do agente

### **Atendimento ao Cliente**

- **RF-10:** Responder automaticamente com base no cadastro
- **RF-11:** Cliente pode solicitar cardápio completo
- **RF-12:** Cliente consulta preço de produto específico
- **RF-13:** Agente responde perguntas comuns
- **RF-14:** Suporte a linguagem natural

### **Administração do Sistema**

- **RF-15:** Login e autenticação segura
- **RF-16:** Dashboard com métricas básicas
- **RF-17:** Suporte a múltiplos estabelecimentos (futuro)

### **Integrações**

- **RF-18:** Integração com API oficial do WhatsApp Business
  - **RF-19:** Armazenar logs de conversas (últimas 50 interações)
  - **RF-20:** Exportar produtos em CSV/Excel
- 

## **6. Requisitos Não Funcionais**

### **Desempenho**

- **RNF-01:** Resposta em até 3s
- **RNF-02:** Suportar 500 produtos sem perda de desempenho
- **RNF-03:** Suportar 100 atendimentos simultâneos

### **Segurança**

- **RNF-04:** Login com senha forte
- **RNF-05:** Dados sensíveis criptografados
- **RNF-06:** HTTPS/TLS 1.2+ obrigatório
- **RNF-07:** Logs acessíveis só a autorizados

## Usabilidade

- **RNF-08:** Painel responsivo (desktop e mobile)
- **RNF-09:** Suporte a linguagem natural
- **RNF-10:** Cadastro/edição em até 3 cliques

## Escalabilidade

- **RNF-11:** Suporte a containers (Docker) e bancos relacionais
- **RNF-12:** Preparado para integrações futuras (pagamentos, delivery)
- **RNF-13:** Multi-estabelecimento (versões futuras)

## Manutenibilidade

- **RNF-14:** Código versionado em GitHub
  - **RNF-15:** Documentação básica de APIs e setup
  - **RNF-16:** Testes automatizados cobrindo 30% no MVP
- 

## 7. Escopo do MVP

### Inclusões (Primeira versão)

- Cadastro e gestão de produtos (nome, descrição, preço, categoria, disponibilidade)
- Painel administrativo responsivo (login, autenticação, CRUD de produtos)
- Integração com API oficial do WhatsApp Business
- Agente respondendo automaticamente com base no cardápio cadastrado
- Configuração básica do agente (mensagem de boas-vindas e respostas padrão)
- Logs de até 50 interações por cliente

### Exclusões (fora do MVP / futuras versões)

- Pagamentos online integrados
- Sistema de delivery com rastreamento de pedidos
- Programa de fidelidade e cupons
- Dashboard avançado de métricas

- Suporte multi-estabelecimento
- 

## 8. Roadmap Futuro

### Versão 1.1 (Curto Prazo)

- Integração com pagamentos online (Pix, Cartão)
- Dashboard simples de métricas (número de interações, produtos mais consultados)
- Suporte a fotos/imagens nos produtos exibidos pelo agente

### Versão 1.2 (Médio Prazo)

- Sistema de delivery integrado (status de pedido: em preparo, a caminho, entregue)
- Histórico de conversas por cliente para personalização de atendimento
- Módulo de promoções e cupons de desconto

### Versão 2.0 (Longo Prazo)

- Programa de fidelidade (pontos, cashback)
  - Suporte multi-estabelecimento (vários restaurantes usando a mesma plataforma)
  - Integração com marketplaces externos (ex: iFood, Rappi) para centralizar gestão
  - Painel avançado de analytics (dashboard com gráficos e relatórios exportáveis)
- 

## 9. Arquitetura e Stack Tecnológico

### Backend

- **Framework:** NestJS com TypeScript
- **ORM:** Prisma ORM
- **Banco de Dados:** PostgreSQL (principal) + Redis (cache/sessões)
- **API:** RESTful com Swagger (decorators nativos do NestJS)
- **Autenticação:** JWT com Guards do NestJS + Passport
- **Validação:** Class-validator e Class-transformer
- **Upload de Arquivos:** AWS S3 ou Google Cloud Storage com Multer
- **Testing:** Jest (integrado ao NestJS)
- **Queue System:** Bull MQ + Redis para processamento assíncrono

### Frontend (Painel Administrativo)

- **Framework:** Next.js 14+ com TypeScript

- **Styling:** Tailwind CSS + Shadcn/ui
- **Autenticação:** NextAuth.js com JWT
- **Estado:** Zustand ou React Query (TanStack Query)
- **Formulários:** React Hook Form + Zod validation
- **Upload:** Upload de imagens com preview
- **Deploy:** Vercel ou AWS Amplify

## Integrações

- **WhatsApp:** WhatsApp Business API oficial
  - **AI/NLP:** OpenAI GPT-4 ou Google Dialogflow
  - **Infraestrutura:** Docker + Kubernetes ou AWS ECS
  - **Monitoramento:** Sentry (erros) + New Relic (performance)
- 

## 10. Critérios de Aceitação

### CA-01: Cadastro de Produtos

**Dado** que o administrador está logado no painel

**Quando** ele preenche o formulário de produto com dados válidos

**Então** o produto deve ser salvo com sucesso e aparecer na lista

**E** estar disponível para consulta no WhatsApp em até 30 segundos

### CA-02: Resposta Automática do Agente

**Dado** que um cliente envia mensagem no WhatsApp

**Quando** solicita o cardápio ou pergunta sobre um produto

**Então** o agente deve responder em até 3 segundos

**E** a resposta deve ser baseada nos produtos cadastrados e disponíveis

### CA-03: Configuração do Agente

**Dado** que o administrador acessa as configurações

**Quando** altera a mensagem de boas-vindas ou tom de voz

**Então** as mudanças devem ser aplicadas imediatamente

**E** refletir nas próximas interações do WhatsApp

### CA-04: Disponibilidade de Produtos

**Dado** que um produto está marcado como indisponível

**Quando** cliente pergunta sobre ele no WhatsApp

**Então** o agente deve informar que está temporariamente indisponível

**E** sugerir produtos similares da mesma categoria

---

## 11. Métricas de Sucesso

### Métricas de Negócio

- **Taxa de Resposta:** 95% das mensagens respondidas em até 3s
- **Conversão:** Aumento de 30% em pedidos comparado ao atendimento manual
- **Satisfação:** NPS > 8 baseado em pesquisa pós-atendimento
- **Retenção:** 80% dos estabelecimentos continuam usando após 3 meses

### Métricas Técnicas

- **Uptime:** 99.5% de disponibilidade
- **Latência:** Tempo médio de resposta < 2s
- **Escalabilidade:** Suportar 100 atendimentos simultâneos sem degradação
- **Precisão do Agente:** 90% de respostas corretas sobre produtos

### Métricas de Produto

- **Adoção:** 50 estabelecimentos cadastrados em 6 meses
  - **Uso:** Média de 20 produtos cadastrados por estabelecimento
  - **Engajamento:** 200 interações por estabelecimento/mês
- 

## 12. Personas e Casos de Uso

### Persona 1: Dona Maria - Proprietária de Lanchonete

**Perfil:** 45 anos, ensino médio, pouca familiaridade com tecnologia

**Dores:** Perde vendas por não conseguir responder WhatsApp rapidamente

**Objetivos:** Automatizar atendimento sem perder qualidade

**Caso de Uso:** Cadastra 15 lanches, configura mensagem acolhedora, monitora vendas

### Persona 2: João - Gerente de Pizzaria

**Perfil:** 32 anos, superior completo, tecnófilo

**Objetivos:** Otimizar operação e aumentar conversão

**Caso de Uso:** Usa métricas para otimizar cardápio, testa diferentes configurações do agente

### Persona 3: Cliente Final - Carlos

**Perfil:** 28 anos, usa WhatsApp diariamente

**Expectativas:** Resposta rápida, informações precisas, processo simples

**Jornada:** Pergunta sobre promoções → Consulta preços → Faz pedido por telefone

### 13. Riscos e Mitigações

#### Riscos Técnicos

Risco	Probabilidade	Impacto	Mitigação
API WhatsApp indisponível	Média	Alto	Implementar fallback e retry automático
Sobrecarga no banco	Baixa	Alto	Cache Redis + otimização de queries
Falha no processamento NLP	Média	Médio	Respostas padrão quando IA falha

#### Riscos de Negócio

Risco	Probabilidade	Impacto	Mitigação
Baixa adoção inicial	Alta	Alto	Período gratuito + suporte ativo
Competição de gigantes	Média	Alto	Foco em nicho específico e relacionamento
Mudanças na política WhatsApp	Baixa	Muito Alto	Monitorar updates + plano B com Telegram

### 14. Premissas e Dependências

#### Premissas

- WhatsApp Business API continuará disponível para desenvolvedores
- Estabelecimentos têm smartphones Android/iOS básicos
- Clientes já usam WhatsApp como canal principal de comunicação
- Donos de negócio conseguem dedicar 2h para setup inicial

#### Dependências Externas

- Aprovação WhatsApp Business:** Necessária para API oficial
- Parceiro de Pagamento:** Integração com gateway (Stripe, PagSeguro)
- Serviço de IA:** OpenAI ou Google para processamento de linguagem natural
- Infraestrutura Cloud:** AWS, GCP ou Azure para hospedagem

#### Dependências Internas

- Equipe de Design:** Protótipos do painel administrativo
- Equipe de DevOps:** Setup de infraestrutura e CI/CD



- **Compliance:** Adequação à LGPD para dados de clientes
- 

## 15. Cronograma de Desenvolvimento com AgentOS

### Metodologia de Desenvolvimento

O desenvolvimento será gerenciado através do **AgentOS**, utilizando agentes especializados para cada área do projeto, garantindo eficiência, qualidade e coordenação entre as diferentes frentes de trabalho.

### Agentes de Desenvolvimento

#### Agent Backend 🛠️

- **Responsabilidade:** Desenvolvimento da API NestJS + Prisma
- **Tarefas:** Endpoints, autenticação, integração WhatsApp, banco de dados
- **Deliverables:** APIs funcionais, documentação Swagger, testes unitários

#### Agent Frontend 🎨

- **Responsabilidade:** Desenvolvimento do painel NextJS
- **Tarefas:** Interfaces, componentes, integração com APIs, responsividade
- **Deliverables:** Painel administrativo completo, componentes reutilizáveis

#### Agent DevOps 🚀

- **Responsabilidade:** Infraestrutura, deploy, monitoramento
- **Tarefas:** Docker, CI/CD, cloud setup, databases, monitoring
- **Deliverables:** Ambiente de produção, pipelines automatizados

#### Agent QA 🧪

- **Responsabilidade:** Qualidade e testes
- **Tarefas:** Testes automatizados, testes manuais, validação de requisitos
- **Deliverables:** Suíte de testes, relatórios de qualidade

#### Agent Integration 🔗

- **Responsabilidade:** Integrações externas
- **Tarefas:** WhatsApp Business API, pagamentos, IA/NLP
- **Deliverables:** Integrações funcionais, documentação de APIs

### Coordenação via AgentOS

### Sprint Planning Automatizado

- AgentOS analisa backlog e distribui tarefas por especialidade
- Cada agente recebe tasks específicas da sua área
- Dependências entre agentes são mapeadas automaticamente
- Timeline otimizada com base na capacidade de cada agente

### **Daily Sync Inteligente**

- Agentes reportam progresso automaticamente
- Identificação de blockers entre teams
- Realocação dinâmica de recursos quando necessário
- Alertas proativos sobre riscos de deadline

### **Code Review Distribuído**

- Agent Backend revisa código de API
- Agent Frontend revisa interfaces e UX
- Agent DevOps valida práticas de infraestrutura
- Agent QA verifica cobertura de testes

## **Sprint 1-2 (Semanas 1-4): Fundação**

**Coordenador:** Agent DevOps + Agent Backend

- **Agent DevOps:** Setup inicial de infraestrutura
  - Configuração Docker + databases
  - CI/CD básico
  - Ambientes dev/staging
- **Agent Backend:** Base da aplicação
  - Setup NestJS + Prisma
  - Autenticação JWT
  - CRUD básico de produtos
  - Migrations iniciais

## **Sprint 3-4 (Semanas 5-8): Interfaces**

**Coordenador:** Agent Frontend + Agent Backend

- **Agent Frontend:** Painel administrativo
  - Setup NextJS + Tailwind
  - Telas de login e dashboard

- CRUD de produtos (UI)
- Integração com APIs
- **Agent Backend:** APIs para frontend
  - Endpoints para painel
  - Upload de imagens
  - Validações de dados

## **Sprint 5-6 (Semanas 9-12): WhatsApp Integration**

**Coordenador:** Agent Integration + Agent Backend

- **Agent Integration:** WhatsApp Business API
  - Setup da API oficial
  - Webhook configuration
  - Message processing
- **Agent Backend:** Lógica do chatbot
  - Processamento de mensagens
  - Consultas ao cardápio
  - Logs de conversas
- **Agent QA:** Testes de integração

## **Sprint 7-8 (Semanas 13-16): Refinamento e Deploy**

**Coordenador:** Todos os agentes

- **Agent Integration:** IA/NLP para respostas inteligentes
- **Agent Frontend:** Configurações do agente, métricas
- **Agent Backend:** Otimizações de performance
- **Agent DevOps:** Deploy produção + monitoramento
- **Agent QA:** Testes end-to-end, UAT

## **Benefícios do AgentOS no Desenvolvimento**

### **Eficiência Máxima**

- Paralelização inteligente de tarefas
- Especialização por domínio técnico
- Redução de conflitos e retrabalho

### **Qualidade Garantida**

- Review automático por especialistas
- Validação contínua de requisitos
- Testes distribuídos por área

### **Adaptabilidade**

- Realocação dinâmica conforme necessidade
- Escalamento de recursos por demanda
- Ajuste de prioridades em tempo real

### **Transparência Total**

- Visibilidade completa do progresso
  - Métricas em tempo real por agente
  - Identificação precoce de riscos
- 

## **16. Definição de Pronto (DoD)**

### **Para Features**

- ☐ Código revisado por pelo menos 1 dev senior
- ☐ Testes unitários com cobertura mínima de 70%
- ☐ Documentação técnica atualizada
- ☐ Testado em ambiente de staging
- ☐ Aprovado pelo PO/PM
- ☐ Sem issues críticos em ferramentas de análise estática

### **Para MVP**

- ☐ Todos os critérios de aceitação atendidos
  - ☐ Performance dentro dos requisitos não-funcionais
  - ☐ Testado com pelo menos 3 estabelecimentos reais
  - ☐ Documentação de usuário completa
  - ☐ Plano de rollback preparado
  - ☐ Monitoramento e alertas configurados
- 

## **17. Aprovações e Sign-off**

### **Stakeholders Necessários**

- **Product Owner:** [Nome] - Responsável pela visão de produto

- **Tech Lead:** [Nome] - Viabilidade técnica e arquitetura
- **Design Lead:** [Nome] - Experiência do usuário
- **DevOps Lead:** [Nome] - Infraestrutura e operações

**Critérios para Aprovação**

- Orçamento aprovado: R\$ [valor] para desenvolvimento
  - Timeline aceita: 16 semanas para MVP
  - Recursos alocados: 4 desenvolvedores + 1 designer + 1 PM
  - Riscos mapeados e planos de mitigação aceitos
- 

**Próximos Passos:**

1. Aprovação formal deste PRD
  2. Definição de orçamento e timeline final
  3. Montagem da equipe de desenvolvimento
  4. Kickoff do projeto e início da Sprint 1
- 

*Documento versão 1.0 - Última atualização: [Data]*  
*Responsável: [Nome do PM/PO]*