

# FASE 4 - Integrações Externas e Pagamentos

---

**Projeto:** DELIVEREI

**Data de Início Planejada:** 09 de Outubro de 2025

**Status:**  Planejamento

---

## Visão Geral

---

A FASE 4 foca em integrações externas essenciais para operação completa do sistema, incluindo:

- WhatsApp Business API para notificações e pedidos
  - Stripe para assinaturas do aplicativo
  - Asaas para pagamentos das lojas
  - Sistema de webhooks para status de pagamento
  - Notificações em tempo real
- 

## Objetivos Principais

---

### 1. Comunicação via WhatsApp

- Notificações automáticas de pedidos
- Confirmação de pedidos via WhatsApp
- Status de entrega em tempo real
- Suporte ao cliente via chat

### 2. Sistema de Pagamentos

- Assinaturas mensais/anuais (Stripe)
- Pagamentos de pedidos (Asaas)
- Gestão de planos e cobranças
- Webhooks de confirmação

### 3. Automação e Webhooks

- Webhooks Stripe (assinaturas)
  - Webhooks Asaas (pagamentos)
  - Processamento assíncrono
  - Logs e auditoria
-

## Arquitetura Proposta

### Backend (NestJS)

```
backend/src/
├── integracoes/
│   ├── whatsapp/
│   │   ├── whatsapp.module.ts
│   │   ├── whatsapp.service.ts
│   │   ├── whatsapp.controller.ts
│   │   └── dto/
│   │       ├── enviar-mensagem.dto.ts
│   │       └── webhook-whatsapp.dto.ts
│   ├── stripe/
│   │   ├── stripe.module.ts
│   │   ├── stripe.service.ts
│   │   ├── stripe.controller.ts
│   │   └── dto/
│   │       ├── criar-assinatura.dto.ts
│   │       └── webhook-stripe.dto.ts
│   └── asaas/
│       ├── asaas.module.ts
│       ├── asaas.service.ts
│       ├── asaas.controller.ts
│       └── dto/
│           ├── criar-cobranca.dto.ts
│           └── webhook-asaas.dto.ts
├── assinaturas/
│   ├── assinaturas.module.ts
│   ├── assinaturas.service.ts
│   ├── assinaturas.controller.ts
│   └── dto/
└── webhooks/
    ├── webhooks.module.ts
    ├── webhooks.service.ts
    └── webhooks.controller.ts
```

## Integrações Detalhadas

### 1. WhatsApp Business API

#### Provedor Sugerido

- **Opção 1:** Twilio WhatsApp API
- **Opção 2:** Meta WhatsApp Business API
- **Opção 3:** Evolution API (self-hosted)

## Funcionalidades

### Notificações Automáticas

```
// Eventos que disparam mensagens WhatsApp
- Novo pedido criado
- Pedido confirmado
- Pedido em preparo
- Pedido saiu para entrega
- Pedido entregue
- Pedido cancelado
```

### Templates de Mensagens

```
📦 *Novo Pedido #{{numero}}*

Olá {{nome_cliente}}!

Seu pedido foi recebido com sucesso!

*Itens:*
{{lista_itens}}

*Total:* R$ {{total}}
*Endereço:* {{endereco}}

Acompanhe seu pedido em: {{link}}

---

✅ *Pedido Confirmado #{{numero}}*

Seu pedido foi confirmado e está sendo preparado!

Previsão de entrega: {{previsao}}

---

🚚 *Pedido Saiu para Entrega #{{numero}}*

Seu pedido está a caminho!

Entregador: {{nome_entregador}}
Previsão: {{previsao}}

---

🎉 *Pedido Entregue #{{numero}}*

Seu pedido foi entregue!

Obrigado por escolher {{nome_empresa}}!

Avalie sua experiência: {{link_avaliacao}}
```

## Endpoints API

```
// POST /api/whatsapp/enviar
{
  "telefone": "+5511999999999",
  "mensagem": "Texto da mensagem",
  "template": "novo_pedido", // opcional
  "parametros": { ... } // opcional
}

// POST /api/whatsapp/webhook
// Recebe mensagens e status de entrega

// GET /api/whatsapp/status/:telefone
// Verifica se número está no WhatsApp
```

## Configuração

```
WHATSAPP_API_URL=https://api.whatsapp.com
WHATSAPP_API_KEY=your_api_key
WHATSAPP_PHONE_NUMBER_ID=your_phone_id
WHATSAPP_BUSINESS_ACCOUNT_ID=your_account_id
```

---

## 2. Stripe (Assinaturas do App)

### Planos Sugeridos

```
const PLANOS = {
  BASICO: {
    nome: 'Básico',
    preco_mensal: 49.90,
    preco_anual: 499.00, // 2 meses grátis
    recursos: [
      'Até 100 pedidos/mês',
      'Catálogo de produtos',
      'Carrinho e checkout',
      'Notificações WhatsApp',
      'Suporte por email'
    ]
  },
  PROFISSIONAL: {
    nome: 'Profissional',
    preco_mensal: 99.90,
    preco_anual: 999.00,
    recursos: [
      'Pedidos ilimitados',
      'Dashboard completo',
      'Cupons de desconto',
      'Múltiplos usuários',
      'Integração Asaas',
      'Suporte prioritário'
    ]
  },
  ENTERPRISE: {
    nome: 'Enterprise',
    preco_mensal: 199.90,
    preco_anual: 1999.00,
    recursos: [
      'Tudo do Profissional',
      'White label',
      'API dedicada',
      'Suporte 24/7',
      'Gerente de conta'
    ]
  }
};
```

### Funcionalidades

#### Gestão de Assinaturas

- Criar assinatura
- Atualizar plano (upgrade/downgrade)
- Cancelar assinatura
- Renovação automática
- Período de teste (trial)

## Endpoints API

```
// POST /api/assinaturas/criar
{
  "empresaId": "uuid",
  "plano": "PROFISSIONAL",
  "periodo": "MENSAL", // ou ANUAL
  "metodoPagamento": "card_token"
}

// GET /api/assinaturas/empresa/:empresaId
// Retorna assinatura ativa

// PATCH /api/assinaturas/:id/plano
{
  "novoPlano": "ENTERPRISE"
}

// DELETE /api/assinaturas/:id
// Cancela assinatura

// POST /api/webhooks/stripe
// Webhook para eventos Stripe
```

## Eventos Stripe

```
// Eventos importantes
- customer.subscription.created
- customer.subscription.updated
- customer.subscription.deleted
- invoice.payment_succeeded
- invoice.payment_failed
- customer.subscription.trial_will_end
```

## Configuração

```
STRIPE_SECRET_KEY=sk_test_...
STRIPE_PUBLISHABLE_KEY=pk_test_...
STRIPE_WEBHOOK_SECRET=whsec_...
STRIPE_PRICE_ID_BASICO_MENSAL=price_...
STRIPE_PRICE_ID_BASICO_ANUAL=price_...
# ... outros price IDs
```

## 3. Asaas (Pagamentos das Lojas)

### Funcionalidades

#### Cobranças

- PIX (instantâneo)
- Boleto bancário
- Cartão de crédito
- Link de pagamento

#### Gestão de Clientes

- Criar cliente no Asaas

- Atualizar dados
- Histórico de pagamentos

## Endpoints API

```
// POST /api/pagamentos/criar
{
  "pedidoId": "uuid",
  "valor": 100.00,
  "metodoPagamento": "PIX", // PIX, BOLETO, CREDIT_CARD
  "cliente": {
    "nome": "João Silva",
    "cpfCnpj": "12345678900",
    "email": "joao@email.com"
  }
}

// GET /api/pagamentos/:id
// Consulta status do pagamento

// POST /api/webhooks/asaas
// Webhook para eventos Asaas

// POST /api/pagamentos/:id/estornar
// Estorna pagamento
```

## Eventos Asaas

```
// Eventos importantes
- PAYMENT_CREATED
- PAYMENT_CONFIRMED
- PAYMENT_RECEIVED
- PAYMENT_OVERDUE
- PAYMENT_DELETED
- PAYMENT_REFUNDED
```

## Configuração

```
ASAAS_API_KEY=your_api_key
ASAAS_ENVIRONMENT=sandbox # ou production
ASAAS_WEBHOOK_TOKEN=your_webhook_token
```

---

## **Novos Models Prisma**

---



```
// Assinaturas Stripe
model Assinatura {
  id String @id @default(uuid())
  empresaId String @unique
  empresa Empresa @relation(fields: [empresaId], references: [id])

  stripeCustomerId String @unique
  stripeSubscriptionId String @unique
  stripePriceId String

  plano String // BASICO, PROFISSIONAL, ENTERPRISE
  periodo String // MENSAL, ANUAL
  status String // ATIVA, CANCELADA, TRIAL, VENCIDA

  dataInicio DateTime
  dataFim DateTime?
  proximaCobranca DateTime?

  valorMensal Decimal

  criadoEm DateTime @default(now())
  atualizadoEm DateTime @updatedAt

  @@index([empresaId])
  @@index([status])
}

// Pagamentos Asaas
model Pagamento {
  id String @id @default(uuid())
  pedidoId String @unique
  pedido Pedido @relation(fields: [pedidoId], references: [id])

  asaasPaymentId String @unique
  asaasCustomerId String

  valor Decimal
  metodoPagamento String // PIX, BOLETO, CREDIT_CARD
  status String // PENDING, CONFIRMED, RECEIVED, OVERDUE, REFUNDED

  pixQrCode String?
  pixCopyPaste String?
  boletoUrl String?
  invoiceUrl String?

  dataPagamento DateTime?
  dataVencimento DateTime?

  criadoEm DateTime @default(now())
  atualizadoEm DateTime @updatedAt

  @@index([pedidoId])
  @@index([status])
  @@index([asaasPaymentId])
}

// Logs de Webhooks
model WebhookLog {
  id String @id @default(uuid())
  origem String // STRIPE, ASAAS, WHATSAPP
  evento String
  payload Json
}
```

```

processado      Boolean    @default(false)
erro            String?
criadoEm        DateTime   @default(now())

@@index([origem])
@@index([processado])
@@index([criadoEm])
}

// Mensagens WhatsApp
model MensagemWhatsApp {
  id            String      @id @default(uuid())
  empresaId     String
  empresa       Empresa     @relation(fields: [empresaId], references: [id])

  pedidoId      String?
  pedido        Pedido?     @relation(fields: [pedidoId], references: [id])

  telefone      String
  mensagem      String
  template      String?
  status        String      // ENVIADA, ENTREGUE, LIDA, FALHOU

  whatsappId    String?     @unique
  erro          String?

  criadoEm      DateTime     @default(now())
  atualizadoEm   DateTime     @updatedAt

  @@index([empresaId])
  @@index([pedidoId])
  @@index([status])
}









```

## Fluxos de Integração

### Fluxo de Pedido com Pagamento

1. Cliente finaliza pedido no checkout  
↓
2. Backend cria pedido (status: PENDENTE)  
↓
3. Backend cria cobrança no Asaas  
↓
4. Cliente recebe link/QR Code de pagamento  
↓
5. Cliente efetua pagamento  
↓
6. Asaas envia webhook de confirmação  
↓
7. Backend atualiza status do pedido (CONFIRMADO)  
↓
8. Backend envia notificação WhatsApp  
↓
9. Admin recebe pedido no painel

## Fluxo de Assinatura

1. Empresa se cadastra no sistema  

2. Escolhe plano e período  

3. Backend cria customer no Stripe  

4. Backend cria subscription no Stripe  

5. Cliente paga primeira mensalidade  

6. Stripe envia webhook de confirmação  

7. Backend ativa assinatura  

8. Empresa pode usar o sistema  

9. Renovação automática mensal/anual

---

## Testes Necessários

### Testes Unitários

- [ ] Serviços de integração (WhatsApp, Stripe, Asaas)
- [ ] Processamento de webhooks
- [ ] Validação de DTOs
- [ ] Lógica de negócio

### Testes de Integração

- [ ] Fluxo completo de pagamento
- [ ] Fluxo de assinatura
- [ ] Envio de mensagens WhatsApp
- [ ] Processamento de webhooks

### Testes E2E

- [ ] Criar pedido e pagar
  - [ ] Assinar plano
  - [ ] Receber notificações
  - [ ] Cancelar assinatura
-

## Dependências Necessárias

---

### Backend

```
{
  "dependencies": {
    "stripe": "^14.0.0",
    "axios": "^1.6.0",
    "@nestjs/axios": "^3.0.0",
    "twilio": "^4.19.0" // se usar Twilio para WhatsApp
  }
}
```

### Frontend

```
{
  "dependencies": {
    "@stripe/stripe-js": "^2.0.0",
    "@stripe/react-stripe-js": "^2.0.0"
  }
}
```

---

## Interface do Usuário

---

### Páginas Admin

#### 1. Configurações de Integração

- Conectar WhatsApp
- Configurar Asaas
- Gerenciar assinatura Stripe

#### 2. Histórico de Pagamentos

- Lista de todos os pagamentos
- Filtros por status e data
- Detalhes de cada pagamento

#### 3. Mensagens WhatsApp

- Histórico de mensagens enviadas
- Status de entrega
- Reenviar mensagens

### Páginas Cliente

#### 1. Pagamento do Pedido

- Escolher método (PIX, Boleto, Cartão)
- Exibir QR Code PIX
- Link do boleto
- Formulário de cartão

#### 2. Acompanhamento

- Status do pagamento

- Status do pedido
- Notificações recebidas

---

## Segurança

---

### Webhooks

- Validação de assinatura (Stripe)
- Token de autenticação (Asaas)
- IP whitelist
- Rate limiting

### Dados Sensíveis

- Não armazenar dados de cartão
- Criptografar tokens de API
- Logs sem informações sensíveis
- HTTPS obrigatório

### Compliance

- PCI DSS (pagamentos)
- LGPD (dados pessoais)
- Termos de uso WhatsApp Business



## Cronograma Sugerido

---

### Semana 1: Infraestrutura

- [ ] Configurar contas (Stripe, Asaas, WhatsApp)
- [ ] Criar models Prisma
- [ ] Migrations
- [ ] Estrutura de módulos

### Semana 2: Stripe

- [ ] Integração Stripe
- [ ] Gestão de assinaturas
- [ ] Webhooks Stripe
- [ ] Testes

### Semana 3: Asaas

- [ ] Integração Asaas
- [ ] Criação de cobranças
- [ ] Webhooks Asaas
- [ ] Testes

### Semana 4: WhatsApp

- [ ] Integração WhatsApp API

- [ ] Templates de mensagens
- [ ] Envio automático
- [ ] Testes

## Semana 5: Frontend

- [ ] Páginas de pagamento
- [ ] Gestão de assinatura
- [ ] Históricos
- [ ] Testes E2E

## Semana 6: Finalização

- [ ] Testes completos
- [ ] Documentação
- [ ] Deploy
- [ ] Monitoramento



## Custos Estimados

### Serviços Mensais

- **Stripe:** 2.9% + R\$ 0,39 por transação
- **Asaas:**
- PIX: 0,99%
- Boleto: R\$ 3,49
- Cartão: 3,99%
- **WhatsApp Business API:**
- Twilio: ~\$0.005 por mensagem
- Meta: Gratuito até 1000 conversas/mês

### Infraestrutura

- Servidor: R\$ 50-200/mês
- Banco de dados: Incluído no Supabase
- CDN: R\$ 0-50/mês



## Métricas de Sucesso

### KPIs

- Taxa de conversão de pagamentos
- Tempo médio de confirmação
- Taxa de entrega de mensagens WhatsApp
- Churn de assinaturas
- MRR (Monthly Recurring Revenue)

### Monitoramento

- Logs de webhooks

- Erros de integração
  - Latência de APIs
  - Uptime dos serviços
- 

## Riscos e Mitigações

---

### Riscos Técnicos

1. **Falha em webhooks**
  - Mitigação: Sistema de retry e logs
2. **Indisponibilidade de APIs**
  - Mitigação: Circuit breaker e fallbacks
3. **Dados inconsistentes**
  - Mitigação: Transações e reconciliação

### Riscos de Negócio

1. **Custos de transação**
    - Mitigação: Análise de volume e negociação
  2. **Compliance**
    - Mitigação: Consultoria jurídica
  3. **Experiência do usuário**
    - Mitigação: Testes extensivos e feedback
- 

## Documentação Necessária

---

- [ ] Guia de integração Stripe
  - [ ] Guia de integração Asaas
  - [ ] Guia de integração WhatsApp
  - [ ] Manual de webhooks
  - [ ] Troubleshooting
  - [ ] FAQ para usuários
- 

## Critérios de Aceitação

---

### Funcional

- [ ] Criar e gerenciar assinaturas Stripe
- [ ] Processar pagamentos via Asaas (PIX, Boleto, Cartão)
- [ ] Enviar notificações WhatsApp automaticamente
- [ ] Processar webhooks corretamente
- [ ] Interface de pagamento funcional

## Não Funcional

- [ ] Tempo de resposta < 2s
  - [ ] Disponibilidade > 99%
  - [ ] Webhooks processados em < 5s
  - [ ] Mensagens WhatsApp entregues em < 10s
- 



## Recursos Úteis

---

### Documentação Oficial

- [Stripe Docs](https://stripe.com/docs) (https://stripe.com/docs)
- [Asaas Docs](https://docs.asaas.com) (https://docs.asaas.com)
- [WhatsApp Business API](https://developers.facebook.com/docs/whatsapp) (https://developers.facebook.com/docs/whatsapp)
- [Twilio WhatsApp](https://www.twilio.com/docs/whatsapp) (https://www.twilio.com/docs/whatsapp)

### Exemplos e Tutoriais

- [Stripe + NestJS](https://docs.nestjs.com/recipes/stripe) (https://docs.nestjs.com/recipes/stripe)
  - [Webhooks Best Practices](https://stripe.com/docs/webhooks/best-practices) (https://stripe.com/docs/webhooks/best-practices)
- 

**Documento criado em:** 08/10/2025

**Versão:** 1.0

**Status:**  Aguardando aprovação para início

**Próximo passo:** Revisar planejamento e iniciar Semana 1