

Documentação de Arquitetura: Sistema de Pagamentos Multi-Provider

1. Visão Geral

Este documento descreve a arquitetura de um gateway de pagamentos escalável e resiliente, projetado para integrar múltiplos provedores de forma transparente. O sistema foca em desacoplamento, segurança e confiabilidade no processamento de transações financeiras.

2. ARQUITETURA DE ALTO NÍVEL

A solução foi construída utilizando o framework Laravel 11, seguindo os princípios de SOLID e padrões de projeto para sistemas distribuídos.

- **Camada de Abstração:** Implementação de uma interface comum que padroniza as ações de todos os provedores (PaymentProviderInterface).
- **Providers Mock:** Desenvolvimento de serviços que simulam operadoras reais (ex.: Stripe, PayPal), permitindo validar o fluxo sem custos reais.
- **Roteamento de Transações:** Sistema capaz de escolher o provedor baseado em parâmetros da requisição, facilitando a escalabilidade e a gestão de custos.

3. FLUXO DE PAGAMENTO E WEBHOOKS

O fluxo de dados garante que cada etapa seja registrada e confirmada para manter a conciliação financeira:

- **Checkout:** O usuário inicia a transação através de uma rota protegida por token de autenticação.
- **Registro:** A tentativa de pagamento é salva no banco de dados com status inicial pendente.
- **Webhook:** Endpoint público que recebe notificações assíncronas dos provedores para confirmar ou negar o pagamento.

4. RESILIÊNCIA E IDEMPOTÊNCIA

Para assegurar a confiabilidade e evitar erros operacionais:

- **Idempotência:** Implementação de travas de cache baseadas no ID único do evento do webhook, garantindo que o mesmo aviso não gere duplicidade de crédito no banco de dados.
- **Tratamento de Falhas:** Uso de blocos de captura de exceções para garantir que erros técnicos não interrompam o serviço e retornem mensagens claras ao cliente.

5. SEGURANÇA E OBSERVABILIDADE

- **Autenticação:** API protegida por sistema de registro e login com emissão de tokens seguros.
- **PCI Compliance:** Estratégia de tokenização onde as credenciais sensíveis são gerenciadas via variáveis de ambiente, minimizando a exposição de dados críticos.
- **Monitoramento:** Registro de logs para rastreamento de transações, essencial para suporte e auditoria.

6. CONCLUSÃO

A arquitetura apresentada cumpre os requisitos de ser modular e resiliente, permitindo a adição de novos provedores com esforço mínimo e garantindo a integridade dos dados financeiros através de mecanismos de segurança e idempotência.