

Projeto Fintec com Backend Laravel

Introdução:

Laravel 11 API com OAuth 2.0 via Laravel Passport.

- Laravel: Framework PHP líder no Brasil (ecossistema fintech: Nubank, iFood), desenvolvimento 3x mais rápido (Eloquent ORM, Artisan CLI), manutenção low-cost (devs abundantes). Suporta monólito modular evoluindo para microservices (Lumen para leves).
- OAuth 2.0 (Passport): Servidor OAuth2 completo (tokens JWT, refresh, scopes: loans:read, payments:write). Integra nativo com mobile (PKCE para React Native). Substitui Keycloak – zero overhead.
- Performance: Laravel Octane (Swoole/RoadRunner) para 10k req/s por worker, escalável em K8s.
- Manter: Microservices event-driven, PostgreSQL, Kafka (via junges/laravel-kafka).

Objetivos: 1M+ usuários, 99.99% uptime, conformidade LGPD/BCB/Open

Projeto Fintec com Backend Laravel

Finance.1. Diagrama de Alto Nível da Arquitetura

```
graph TB
    subgraph "Mobile App (React Native)"
        A["App iOS/Android<br/>- Cadastro User<br/>- Solicitar Empréstimo<br/>- Visualizar Extrato"]
    end

    subgraph "API Gateway (AWS API Gateway + Kong)"
        G["API Gateway<br/>- Rate Limiting<br/>- OAuth2 Proxy<br/>- Routing"]
    end

    subgraph "Microservices (Laravel 11 + Octane + Docker/K8s)"
        S1["Auth Service<br/>- Laravel Passport (OAuth2)"]
        S2["User Onboarding<br/>- Cadastro + KYC"]
        S3["Open Finance Service<br/>- Consent + Extrato"]
        S4["Loan Service<br/>- Análise + Aprovação"]
        S5["Payment Orchestrator<br/>- Emitir/Receber PIX"]
    end

    subgraph "Infra Compartilhada"
        DB["(PostgreSQL<br/>- Master/Slaves<br/>- Sharding por User ID)"]
        C["Redis<br/>- Cache + Sessions + Queues"]
        Q["Kafka<br/>- Events: LoanRequested, PaymentProcessed<br/>- Laravel Kafka"]
        ES["Elasticsearch<br/>- Logs + Analytics"]
    end

    subgraph "Externals"
        E1["Receita Federal API (CNPJ)"]
        E2["Validação Docs API (OCR/CNHRG)"]
        E3["Open Finance DIIs (BCB)"]
        E4["PSP: PagSeguro/STC (PIX)"]
    end

    subgraph "Monitoring & Resilience"
        M["Prometheus + Grafana + Telescope<br/>- Alerts PagerDuty"]
        CB["Circuit Breaker<br/>- Laravel Retryable + Resilience4j-like"]
    end

    A --> G
    G --> S1
    G --> S2
    G --> S3
    G --> S4
    G --> S5

    S2 -. Async .-> Q
    S3 --> E3
    S4 --> Q
    S5 <--> E4
    S2 --> E1
    S2 --> E2

    S1 --> DB
    S2 --> DB
    S3 --> DB
    S4 --> DB
    S5 --> DB

    S1 --> C
    S2 --> C
    S4 --> C

    Q --> S5
    Q --> ES

    S2 -. Retry/CB .-> E1
    S3 -. Retry/CB .-> E3

    style DB fill:#f9f,stroke:#333
    style Q fill:#ff9,stroke:#333
```

Projeto Fintec com Backend Laravel

Fluxo OAuth2 Exemplo:

1. Mobile: POST /oauth/token (client_id/secret + PKCE) → Access Token (1h) + Refresh.
2. Requests: Authorization: Bearer {token} → Gateway valida via Introspect.
3. Scopes: user:write (cadastro), loans:request (empréstimos).

1. Justificativas de Tecnologias (Atualizadas)

Componente	Tecnologia	Justificativa
Mobile	React Native	Inalterado. @react-native-oauth para PKCE.
Backend	Laravel 11 + Octane	Rápido MVP (migrations, jobs prontos). Octane: async/high-throughput. PHP 8.3: JIT + Fibers.
Auth	Laravel Passport	OAuth2/OIDC nativo. Personal Access Tokens para mobile. Scopes granulares.
API Gateway	AWS API Gateway + Kong	Proxy OAuth (valida tokens via /oauth/introspect).
DB	PostgreSQL	Eloquent ORM mapeia DER perfeitamente.
Cache/Queues	Redis + Laravel Horizon	Jobs async (KYC), broadcasting.
Events	Kafka (junges/laravel-kafka)	Orquestração robusta. Fallback: Laravel Queues.
Cloud	AWS (EKS, Vapor opcional)	Laravel Vapor: Serverless deploy (zero K8s ops).

2. Diagrama de Entidade-Relacionamento (DER) Inalterado – Laravel Migrations geram automaticamente:

Relacionamentos via Eloquent: hasMany, belongsTo. 4. Plano de Escalabilidade Horizontal: PHP-FPM/Octane workers em K8s HPA. Supervisor gerencia processes.

- DB: Inalterado (sharding via user_id % N em Eloquent).
- Queues: Horizon auto-scales workers. Kafka producers/consumers.
- Tráfego: ALB + Laravel RateLimiter (Redis-based).
- Otimização: OpCache + RoadRunner → <20ms latency.

3. Plano de Resiliência e Monitoramento

- Falhas:

Falha	Mitigação
Terceiros	Circuit Breaker: spatie/laravel-circuit-breaker ou custom (Redis state). Retry: withBackoffRetries().
Duplicados	Idempotency: unique_id em DB (upsert).

- Failover/Backup: Inalterado.
- Monitoramento:

Tool	Uso
Laravel Telescope	Debug queues/jobs/errors.
Prometheus Exporter	Métricas Octane (via blackfire/prometheus-exporter).
Sentry	Erros reais-time.

4. Padrões de Desenvolvimento (Atualizados)

- Versionamento: GitFlow.
- Code Quality:

Prática	Tool
Revisão	GitHub PRs + Laravel Pint (style).
Testes	Pest/PHPUnit: 80% coverage. Dusk (E2E), Horizon Tests.
CI/CD	GitHub Actions: php artisan test → Deploy Vapor (serverless) ou ArgoCD. Blue/Green via Envoy.

- Padrões: API Resources (JSON:API), Filament (admin), Domain Events.
- Segurança: Sanctum (SPA fallback), Laravel Shield.

Comandos Iniciais:

```
composer create-project laravel/laravel fintech-api
```

```
composer require laravel/passport junges/laravel-kafka
```

```
php artisan passport:install
```

```
php artisan octane:start --server=swoole
```

Bônus: Diferenciais (Inalterados + Laravel Boost)

- Multi-país: Laravel Localization + multi-tenant DB.
- Vapor: Zero servers, auto-scale global.
- Circuit Breaker: Nativo em jobs: CircuitBreaker::run().

Impacto da Mudança de Java para Laravel:

Métrica	Spring Boot	Laravel
Time-to-MVP	4 semanas	2 semanas
Dev Cost (BR)	Alto	Baixo
Throughput	15k req/s	12k req/s (Octane)
Ops	K8s pesado	Vapor leve