

# Manual t cnico â PagosDigitales

## 1. Prop sito

Este manual t cnico describe la operaci n interna del servicio PagosDigitales, su arquitectura, configuraci n, integraci n con proveedores y gu as de despliegue y operaci n.

2. Alcance â Aplicaci n Spring Boot 3.x con Apache Camel. â API REST para pagos, bancos y notificaciones. â Confirmaciones SafetyPay con firma SHA-256. â Configuraci n por proveedor en application.yaml.

## 3. Arquitectura

Estilo: monolito en capas (API, Application, Domain, Infrastructure) con integraci n din mica por proveedor.

Componentes clave: â Controllers: expone endpoints REST. â Mappers: transforman requests/responses hacia el formato de cada proveedor. â Servicios de dominio: firma y resoluci n de proveedor. â Camel Routes: enrutan solicitudes hacia proveedores externos.

4. Estructura de paquetes â com.femsa.gpf.pagosdigitales.api: controladores y DTOs. â com.femsa.gpf.pagosdigitales.application: orquestaci n y mapeos. â com.femsa.gpf.pagosdigitales.domain: l gica de negocio. â com.femsa.gpf.pagosdigitales.infrastructure: rutas Camel y configuraci n.

5. Endpoints M todo | Ruta | Descripci n | Content-Type | Respuesta --- | --- | --- | ---  
| --- POST | /api/v1/banks | Consulta bancos por proveedor o todos | JSON | JSON GET |  
/api/v1/payments | Consulta pagos por operationid | JSON (body) | JSON POST |  
/api/v1/direct-online-payment-requests | Solicitud de pago en l nea | JSON | JSON POST |  
/api/v1/payments/notifications/merchant-events | Notificaciones de eventos del comercio | JSON | JSON POST |  
/api/v1/safetypay/confirmation | Webhook SafetyPay (CSV firmado) | application/x-www-form-urlencoded | text/plain GET | /api/v1/pagos/test | Health check | â | text/plain

6.1 Pagos en l nea 1. Controller valida proveedor y solicitud. 2. Mapper crea payload del proveedor seg n mapping. 3. Camel Route arma URL y headers din micamente. 4. Respuesta normalizada hacia DTO interno.

6.2 Consulta de pagos 1. Controller valida paymentprovidercode y operationid. 2. Camel Route agrega par metros de consulta. 3. Respuesta normalizada a DTO interno.

6.3 Consulta de bancos 1. Controller consulta proveedor espec fico o todos. 2. Camel Route construye URL con par metros. 3. Respuesta normalizada por mapping.

6.4 Merchant Events 1. Controller valida proveedor. 2. Mapper transforma payload. 3. Camel Route envía notificación.

6.5 SafetyPay Confirmation 1. Controller mapea form-urlencoded a DTO. 2. Service valida API Key, firma e IP permitida. 3. Aplica idempotencia y registra notificación. 4. Responde CSV firmado.

## 7. Configuración

Ubicación: src/main/resources/application.yaml

7.1 Base de datos de server.port de spring.datasource.\* de spring.jpa.\*

7.2 Proveedores por endpoint de direct-online-payment-requests.providers.\* de merchant-events.providers.\* de payments.providers.\* de getbanks.providers.\*

7.3 Mappings de direct-online-payment-requests.mapping.\* de merchant-events.mapping.\* de payments.mapping.\* de getbanks.mapping.\*

7.4 Códigos y errores de providers-pay.codes de error-mapping.providers.\*

7.5 SafetyPay de safetypay.confirmation.enabled de safetypay.confirmation.providers.\*

8. Seguridad de Validación de firma SHA-256 y API Key en SafetyPay. de Lista opcional de IPs permitidas para confirmaciones. de Headers X-Api-Key y X-Version por proveedor configurados en YAML.

## 9. Despliegue

9.1 Requisitos de Java 17 de Maven de Oracle JDBC accesible

9.2 Build bash mvn clean package

9.3 Docker (referencial) dockerfile FROM eclipse-temurin:17-jre WORKDIR /app COPY target/pagosdigitales-0.0.1-SNAPSHOT.jar app.jar EXPOSE 8080 ENTRYPOINT ["java", "-jar", "app.jar"]

10. Operación y monitoreo de Revisar logs de Camel para trazabilidad de requests/response. de Validar métricas de latencia por endpoint. de Ajustar timeouts por proveedor según SLA.

## 11. Ejemplos de uso

11.1 SafetyPay Confirmation bash curl -X POST

http://localhost:8080/api/v1/safetypay/confirmation \ -H "Content-Type: application/x-www-form-urlencoded" \ --data-urlencode "ApiKey=MIAPIKEY" \ --data-urlencode

```
"paymentprovidercode=235689" \ --data-urlencode "RequestDateTime=2020-10-20T12:27:27" \
--data-urlencode "MerchantSalesID=Prueba1" \ --data-urlencode "ReferenceNo=Ref1" \
--data-urlencode "CreationDateTime=2020-10-20T12:27:27" \ --data-urlencode "Amount=50.00"
\ --data-urlencode "CurrencyID=USD" \ --data-urlencode "PaymentReferenceNo=606973" \
--data-urlencode "Status=102" \ --data-urlencode "Signature=FIRMACALCULADA"
```

#### 11.2 Direct Online Payment Requests bash curl -X POST

```
http://localhost:8080/api/v1/direct-online-payment-requests \ -H "Content-Type:
application/json" \ -d '{ "chain": 1, "store": 148, "storename": "FYBECA AMAZONAS", "pos":
90, "paymentprovidercode": 235689, "salesamount": { "value": 50.00, "currencycode": "USD"
}, "countrycode": "EC", "bankid": "0123", "merchantsetpayamount": true,
"expirationtimeminutes": 30, "languagecode": "ES", "custommerchanname": "TIENDA",
"merchantsalesid": "Prueba1", "requestedpaymenttype": "paycash", "transactionemail":
"cliente@correo.com", "sendemailshopper": true }'
```