CVDex

Appendix_OpenAPI

Finanzas SD – Architecture, Flows & SOPs

Arquitectura, Flujos y Procedimientos

November 10, 2025

1 Appendix: OpenAPI Specification / Apéndice: Especificación OpenAPI

1.1 EN: API Contract Reference

1.1.1 Overview

This appendix provides a reference to the OpenAPI specification for the Finanzas SD API. The complete specification is maintained in the `openapi/` directory of this repository.

1.1.2 API Specification Location

The full OpenAPI 3.0 specification is available at:

- **File**: `openapi/finanzas.yaml`
- **Format**: YAML (OpenAPI 3.0)
- **Validation**: Automated via Spectral linting

1.1.3 Key API Endpoints

Authentication

- `POST /auth/login` - User login
- `POST /auth/logout` - User logout
- `POST /auth/refresh` - Refresh access token
- `GET /auth/me` - Get current user profile

Pre-facturas

- `GET /api/prefacturas` - List pre-facturas
- `POST /api/prefacturas` - Create pre-factura
- `GET /api/prefacturas/{id}` - Get pre-factura details
- `PUT /api/prefacturas/{id}` - Update pre-factura
- `DELETE /api/prefacturas/{id}` - Delete pre-factura (draft only)
- `POST /api/prefacturas/{id}/approve` - Approve pre-factura
- `POST /api/prefacturas/{id}/reject` - Reject pre-factura
- `GET /api/prefacturas/{id}/pdf` - Download pre-factura PDF

Budgets

- `GET /api/budgets` - List budgets
- `POST /api/budgets` - Create budget
- `GET /api/budgets/{id}` - Get budget details

- PUT /api/budgets/{id} - Update budget
- GET /api/budgets/{id}/utilization - Get budget utilization

Projects

- GET /api/projects - List projects
- POST /api/projects - Create project
- GET /api/projects/{id} - Get project details
- PUT /api/projects/{id} - Update project
- GET /api/projects/{id}/prefacturas - Get project pre-facturas
- GET /api/projects/{id}/budget-summary - Get project budget summary

Reports

- POST /api/reports/generate - Generate report (PDF/CSV)
- GET /api/reports/{id} - Get report status
- GET /api/reports/{id}/download - Download generated report

Audit

- GET /api/audit/logs - Get audit logs
- GET /api/audit/logs/{id} - Get specific audit log entry

Notifications

- GET /api/notifications - Get user notifications
- PUT /api/notifications/{id}/read - Mark notification as read
- DELETE /api/notifications/{id} - Delete notification

1.1.4 Authentication & Authorization

Authentication Flow

1. User authenticates via AWS Cognito
2. Client receives JWT access token and refresh token
3. Client includes access token in Authorization header for API requests
4. API Gateway validates token with Cognito
5. Lambda function receives user context from verified token

Authorization Flow

1. API request validated and authenticated
2. Lambda function extracts user identity and requested action

3. AVP policy evaluation called with user attributes and resource attributes
4. AVP returns ALLOW or DENY decision
5. Lambda proceeds or returns 403 Forbidden

Required Headers

```
Authorization: Bearer <access_token>
Content-Type: application/json
```

1.1.5 Common Request/Response Patterns

[] { "success": **true**, "data": { "id": "pf-2024-001", "status": "approved", "...": "..." }, "timestamp": "2024-11-10T06:45:00Z" }

Success Response

[] { "success": **false**, "error": { "code": "VALIDATION_ERROR", "message": "Amount exceeds remaining budget", "details": { "field": "amount", "requested": 5000, "available": 3000 } }, "timestamp": "2024-11-10T06:45:00Z" }

Error Response

1.1.6 Error Codes

Client Errors (4xx)

- `400` - Bad Request (validation error)
- `401` - Unauthorized (authentication required)
- `403` - Forbidden (insufficient permissions)
- `404` - Not Found (resource does not exist)
- `409` - Conflict (resource state conflict)
- `422` - Unprocessable Entity (semantic error)
- `429` - Too Many Requests (rate limit exceeded)

Server Errors (5xx)

- `500` - Internal Server Error
- `502` - Bad Gateway
- `503` - Service Unavailable
- `504` - Gateway Timeout

1.1.7 Rate Limiting

- **Default Rate**: 100 requests per minute per user
- **Burst Capacity**: 200 requests
- **Report Generation**: 10 requests per hour per user
- **PDF Download**: 50 requests per hour per user

1.1.8 Pagination

List endpoints support pagination via query parameters:

- page - Page number (default: 1)
- `limit` - Items per page (default: 20, max: 100)

Response includes pagination metadata:

[] { ”data”: [...], ”pagination”: { ”page”: 1, ”limit”: 20, ”total”: 150, ”totalPages”: 8, ”hasNext”: **true**, ”hasPrev”: **false** } }

1.1.9 Filtering and Sorting

List endpoints support filtering and sorting:

- `filter[field]=value` - Filter by field value
- `filter[status]=approved,pending` - Multiple values (OR)
- `sort=field` - Sort ascending
- `sort=-field` - Sort descending

Example:

```
GET /api/prefacturas?filter[status]=pending&sort=-createdAt&limit=50
```

1.1.10 Postman Collection

A Postman collection with example requests is available at:

- **File**: `postman/Finanzas-SD.postman_collection.json`
- **Environment**: `postman/Finanzas-SD.postman_environment.json`

1.1.11 Testing the API

Automated API tests are available:

- **Contract Tests**: `scripts/test-api-routes-complete.sh`

- **Smoke Tests**: `scripts/finanzas-smoke-tests.sh`
- **E2E Tests**: `scripts/finanzas-e2e-smoke.sh`

1.2 ES: Referencia del Contrato de API

1.2.1 Descripción General

Este apéndice proporciona una referencia a la especificación OpenAPI para la API de Finanzas SD.

1.2.2 Ubicación de la Especificación de API

La especificación completa de OpenAPI 3.0 está disponible en:

- **Archivo**: `openapi/finanzas.yaml`
- **Formato**: YAML (OpenAPI 3.0)
- **Validación**: Automatizada a través de linting Spectral

[Traducción de todas las secciones de API]

1.2.3 Endpoints Principales de la API

[Traducción de todos los endpoints]

1.2.4 Autenticación y Autorización

[Traducción de flujos de autenticación y autorización]

1.2.5 Códigos de Error

[Traducción de códigos de error]

1.2.6 Limitación de Tasa

[Traducción de especificaciones de límites]

1.2.7 Colección de Postman

Una colección de Postman con solicitudes de ejemplo está disponible en:

- **Archivo**: `postman/Finanzas-SD.postman_collection.json`
- **Entorno**: `postman/Finanzas-SD.postman_environment.json`