

WS1 Deliverable - OpenAPI Specification & Postman Collection

Finanzas SD – Architecture, Flows & SOPs
Arquitectura, Flujos y Procedimientos

November 11, 2025

1 WS1 Deliverable - OpenAPI Specification & Postman Collection

1.1 Executive Summary

Complete API contract specification delivered for the Finanzas financial planning platform. All requirements met with 100% endpoint coverage, validated OpenAPI 3.1 specification, comprehensive Postman collection, and detailed API documentation.

1.2 Deliverables

1.2.1 1. OpenAPI 3.1 Specification ✓

File: openapi/finanzas.yaml

- **22 Endpoints** across 9 categories
- **24 Complete Schemas** with strict typing
- **Realistic Examples** for all requests/responses
- **Error Handling** (400, 401, 403, 404)
- **Validation Status:** ✓ PASS (Spectral lint - 0 errors)

Categories: 1. Projects (3 endpoints) - Creation, listing, handoff 2. Catalog (3 endpoints) - Budget rubros management 3. Allocations (2 endpoints) - Budget planning and bulk updates 4. Payroll (3 endpoints) - Ingestion, monthly closing, coverage reports 5. Adjustments (2 endpoints) - Budget adjustments with pro-rata 6. Movements (5 endpoints) - Financial movements with approval workflow 7. Alerts (1 endpoint) - Financial alerts and notifications 8. Providers (2 endpoints) - Vendor management

1.2.2 2. Postman Collection ✓

File: postman/Finanzas.postman_collection.json

- **22 Mock Endpoints** with complete responses
- **Collection-level Authentication** (Bearer JWT)
- **Environment Variables** (base_url, jwt_token)
- **Organized by Category** (9 folders)
- **Ready for Mock Server** deployment

1.2.3 3. API Documentation ✓

File: docs/api-contracts.md

- **Complete API Overview** with authentication details
- **8 Detailed cURL Examples** with explanations
- **RBAC Documentation** (Cognito JWT groups)

- **Error Response Guide** (all 4xx codes)
- **Pagination & Rate Limiting** documentation
- **Webhook Integration** guide

1.2.4 4. Supporting Documentation ✓

Endpoint Coverage Table (docs/endpoint-coverage.md): - Complete mapping of all 22 endpoints - Validation status for each endpoint - Schema implementation checklist - Special features documentation

Postman Test Report (docs/postman-test-report.md): - Detailed test results for all endpoints - Mock server setup instructions - Quality metrics (100% coverage) - Newman CLI testing guide

Spectral Configuration (.spectral.yaml): - OpenAPI validation ruleset - Used for automated validation

1.3 Technical Specifications

1.3.1 Authentication

- **Method:** AWS Cognito JWT (OIDC)
- **Header:** Authorization: Bearer <token>
- **RBAC Groups:** SDT, PMO, Finance
- **Region:** us-east-2

1.3.2 API Design

- **Spec Version:** OpenAPI 3.1.0
- **Base Path:** /finanzas/
- **Response Format:** JSON
- **Date Format:** ISO 8601
- **ID Pattern:** {type}_{alphanumeric10}

1.3.3 Key Features

1. **Project Handoff:** Budget split between engineers (pct_ingenieros) and SDM (pct_sdm)
2. **Pro-Rata Distribution:** Automatic adjustment distribution across remaining months
3. **Coverage Calculation:** Formula: $(\text{ingresos_facturados} / \text{nomina}) \times 100$
4. **Approval Workflow:** Multi-state approval process for movements
5. **Execution Types:** Support for mensual, puntual, and por_hito rubros

1.4 Validation & Testing

1.4.1 OpenAPI Validation (Spectral)

- ✓ PASS - No errors found
- ✓ All endpoints have complete schemas
- ✓ All examples are valid against schemas
- ✓ All required fields are present
- ✓ Pattern validation passes
- ✓ Enum validation passes

Command Used:

```
[] npx spectral lint openapi/finanzas.yaml --format=stylish
```

1.4.2 Quality Metrics

- **Endpoint Coverage:** 22/22 (100%)
- **Schema Coverage:** 24/24 (100%)
- **Example Coverage:** 22/22 (100%)
- **Error Handling:** 22/22 (100%)
- **Spectral Validation:** ✓ PASS (0 errors)

1.5 Example Endpoints

1.5.1 1. POST /projects - Create Project

Creates new project with \$1.5M budget:

```
[] curl -X POST https://api.finanzas.example.com/finanzas/projects \ -H "Authorization: Bearer <token>" \ -H "Content-Type: application/json" \ -d '{"name":"Mobile Banking App MVP","code":"PROJ-2025-001",...}'
```

Response: 201 Created with complete project object

1.5.2 2. POST /projects/{id}/handoff - Project Handoff

Transfers project with 65/35 budget split:

```
[] curl -X POST .../projects/proj_7x9k2m4n8p/handoff \ -d '{"pct_ingenieros":65,"pct_sdm":35,...}'
```

Response: Budget split calculation (Engineers: \$975K, SDM: \$525K)

1.5.3 3. GET /projects/{id}/plan?mes=2025-10 - Monthly Plan

Retrieves financial plan with breakdown: **Response:** plan_ing, plan_sdm, non_recurrentes, total_mes

1.5.4 4. POST /close-month?mes=2025-10 - Close Month

Closes period and calculates coverage: **Response:** 114.72% coverage, \$15.5K positive gap, alerts

1.5.5 5. POST /adjustments - Budget Adjustment

Creates \$1M adjustment with pro-rata forward distribution: **Response:** Distribution across 3 months (\$333,333.33 each)

1.6 Schema Highlights

1.6.1 Key Data Models

1. **Project:** Core project entity with MOD (Monto Original Dealsheet)
2. **Handoff:** Budget acceptance with percentage splits
3. **Rubro:** Budget line items with execution types (mensual/puntual/por_hito)
4. **PlanMes:** Monthly financial plan with recurrent and non-recurrent costs
5. **CierreMes:** Month closing with coverage calculation and alerts
6. **Adjustment:** Budget adjustment with pro-rata distribution methods
7. **Movement:** Financial movement with approval state machine

1.6.2 Enumerations

- **Project Status:** active, completed, on_hold, cancelled
- **Execution Types:** mensual, puntual, por_hito
- **Movement Types:** gasto, ingreso, ajuste
- **Approval States:** pending, approved, rejected
- **Alert Severities:** critical, warning, info

1.7 Files Delivered

```
openapi/
  └── finanzas.yaml          (OpenAPI 3.1 specification)

postman/
  └── Finanzas.postman_collection.json (Postman collection with mocks)

docs/
```

api-contracts.md	(API documentation with cURL examples)
endpoint-coverage.md	(Coverage verification table)
postman-test-report.md	(Postman testing report)
.spectral.yaml	(OpenAPI validation config)

1.8 Green Criteria Checklist

- ✓ **OpenAPI validates** with no errors (Spectral)
- ✓ **All endpoints covered** (22/22 = 100%)
- ✓ **Examples consistent** and valid against schemas
- ✓ **Postman mocks respond** with 2xx for all requests
- ✓ **Documentation complete** with 8+ cURL examples
- ✓ **Peer review ready** with comprehensive documentation
- ✓ **Endpoint coverage table** included
- ✓ **Test evidence** documented

1.9 Usage Instructions

1.9.1 For Developers

1. Review docs/api-contracts.md for API overview
2. Import openapi/finanzas.yaml into your IDE/tools
3. Use OpenAPI spec to generate server stubs
4. Refer to examples for expected request/response formats

1.9.2 For Testers

1. Import postman/Finanzas.postman_collection.json into Postman
2. Create mock server for testing
3. Use docs/postman-test-report.md for test scenarios
4. Run Newman for automated testing

1.9.3 For Documentation

1. docs/api-contracts.md provides complete API reference
2. Use OpenAPI spec with Swagger UI or Redoc
3. Share Postman collection for interactive documentation

1.10 Next Steps

1. ✓ **PR Created:** contracts/finanzas-openapi-mvp
2. □ **Review:** Technical and peer review of artifacts

3. **Approval:** Green criteria verification
4. **Merge:** Integration into main branch
5. **Deploy:** Mock server and documentation portal

1.11 Contact

- **API Lead:** Service Delivery Team (SDT)
 - **Repository:** valencia94/financial-planning-u
 - **Branch:** contracts/finanzas-openapi-mvp
 - **Region:** us-east-2
-

1.12 Validation Evidence

1.12.1 Spectral Lint Output

```
$ npx spectral lint openapi/finanzas.yaml --format stylish
```

No results with a severity of 'error' found!

✓ All validations passed

1.12.2 Endpoint Verification

Total Endpoints Implemented: 22

- Projects: 3/3 ✓
- Catalog: 3/3 ✓
- Allocations: 2/2 ✓
- Payroll: 3/3 ✓
- Adjustments: 2/2 ✓
- Movements: 5/5 ✓
- Alerts: 1/1 ✓
- Providers: 2/2 ✓
- Webhooks: 1/1 ✓

1.13 Conclusion

WS1 DELIVERABLE COMPLETE

All requirements satisfied: - Complete OpenAPI 3.1 specification - Postman collection with mocks - Comprehensive API documentation - Full validation and testing - Ready for R1 billable delivery

Status: GREEN ✓