

# Financial Data Project

## Documentacion Tecnica

Fecha: 30/01/2026

Version: 3.0.0

Puntuacion Enterprise: 8.3/10

## Indice

- \* 1. Resumen Ejecutivo
- \* 2. Arquitectura del Sistema
- \* 3. Estructura del Proyecto
- \* 4. Instalacion y Configuracion
- \* 5. Modulos Principales
- \* 6. Base de Datos
- \* 7. Seguridad
- \* 8. Testing y Calidad
- \* 9. Mantenimiento
- \* 10. Metricas de Calidad

## 1. Resumen Ejecutivo

Financial Data Project es un sistema de gestion de carteras de inversion que permite el seguimiento diario de posiciones, analisis tecnico, y valoracion de portfolios en multiples divisas.

### Caracteristicas Principales

- \* Dashboard interactivo con Streamlit
- \* Integracion con Yahoo Finance
- \* Soporte multi-divisa (EUR, USD, CAD, GBP, CHF)
- \* Tracking de acciones, ETFs y futuros
- \* Analisis tecnico (RSI, medias moviles)
- \* Gestion de operaciones y efectivo
- \* Autenticacion opcional del dashboard

### Tecnologias Utilizadas

| Componente    | Tecnologia          |
|---------------|---------------------|
| Backend       | Python 3.11+        |
| Base de Datos | SQLite / PostgreSQL |
| ORM           | SQLAlchemy 2.0      |
| Dashboard     | Streamlit 1.28+     |
| Graficos      | Plotly 5.18+        |
| Testing       | Pytest 7.4+         |

## 2. Arquitectura del Sistema

### Capas del Sistema

|                      |   |
|----------------------|---|
| <b>Presentacion:</b> | Dashboard Streamlit con multiples paginas                   |
| <b>Servicios:</b>    | PortfolioService, ExchangeRateService, DailyTrackingService |
| <b>Datos:</b>        | YahooClient, DatabaseManager                                |
| <b>Persistencia:</b> | SQLite/PostgreSQL con SQLAlchemy                            |

### Patrones de Diseno

|                         |                                  |
|-------------------------|----------------------------------|
| <b>Singleton:</b>       | Servicios unicos para eficiencia |
| <b>Context Manager:</b> | Gestion segura de sesiones BD    |
| <b>Service Layer:</b>   | Logica de negocio encapsulada    |

### 3. Estructura del Proyecto

```
financial-data-project/  
  src/                #Codigo principal  
    config.py         #Configuracion  
    database.py       #Modelos y manager  
    portfolio_data.py #Servicio portfolio  
    validators.py     #Validacion  
    technical.py      #Indicadores  
  web/  
    app.py            #Dashboard Streamlit  
  scripts/           #Scripts utilidad  
  tests/             #Suite de tests  
  data/              #Base de datos
```

## 4. Instalacion y Configuracion

### Requisitos

- \* Python 3.11 o superior
- \* pip (gestor de paquetes)

### Pasos de Instalacion

```
# 1. Instalar dependencias
pip install -r requirements.txt

# 2. Copiar configuracion
copy .env.example .env

# 3. Configurar variables en .env

# 4. Ejecutar dashboard
streamlit run web/app.py --server.port 8514
```

### Variables de Entorno

| Variable               | Descripcion                |
|------------------------|----------------------------|
| DATABASE_URL           | URL conexion BD            |
| DASHBOARD_PASSWORD     | Contraseña dashboard       |
| DASHBOARD_AUTH_ENABLED | Activar auth (true/false)  |
| LOG_LEVEL              | Nivel logging (INFO/DEBUG) |

## 5. Modulos Principales

| Modulo                   | Descripcion                                    |
|--------------------------|--|
| config.py                | Gestion de configuracion con pydantic-settings |
| database.py              | Modelos SQLAlchemy y DatabaseManager           |
| portfolio_data.py        | Calculos de valoracion y metricas              |
| exchange_rate_service.py | Tipos de cambio centralizados                  |
| daily_tracking.py        | Gestion de holdings y operaciones              |
| validators.py            | Validacion de entrada                          |
| technical.py             | Indicadores tecnicos (RSI, MA, etc.)           |
| yahoo_client.py          | Cliente Yahoo Finance                          |

## 6. Base de Datos

### Tablas Principales

| Tabla          | Descripcion             |
|----------------|-------------------------|
| exchanges      | Bolsas de valores       |
| symbols        | Simbolos con metadatos  |
| price_history  | Historico precios OHLCV |
| holding_diario | Posiciones diarias      |
| stock_trades   | Historial operaciones   |
| daily_cash     | Saldos efectivo         |
| cash_movements | Movimientos efectivo    |

### Conexion

SQLite: sqlite:///data/finacial\_data.db

PostgreSQL: postgresql://user:pass@host:5432/dbname



## 7. Seguridad

### Medidas Implementadas

- \* SQL Injection: Queries parametrizadas
- \* Validacion: Modulo validators.py
- \* Autenticacion: Password configurable
- \* Secretos: Variables de entorno (.env)
- \* Excepciones: Manejo tipado con logging
- \* Escaneo: Bandit en pre-commit

### Activar Autenticacion

```
# En archivo .env:  
DASHBOARD_AUTH_ENABLED=true  
DASHBOARD_PASSWORD=contrasena_segura
```

## 8. Testing y Calidad

### Suite de Tests

- \* 225 tests automatizados
- \* 9 modulos de test
- \* Fixtures en conftest.py
- \* BD in-memory para aislamiento

### Comandos

```
# Ejecutar tests
pytest tests/ -v

# Con cobertura
pytest tests/ --cov=src
```

### Herramientas de Calidad

- \* Black (formateo)
- \* isort (imports)
- \* Flake8 (linting)
- \* Bandit (seguridad)
- \* Mypy (tipos)
- \* pre-commit (hooks)

## 9. Mantenimiento

### Tareas Periodicas

- \* Descarga automatica de precios
- \* Backup de base de datos
- \* Actualizacion de dependencias
- \* Revision de logs

### Scripts Utiles

| Script              | Funcion                 |
|---------------------|-------------------------|
| batch_download.py   | Descarga masiva precios |
| import_ib_data.py   | Importar datos IB       |
| import_portfolio.py | Importar portfolio      |

## 10. Metricas de Calidad

### Puntuacion Enterprise-Ready

| Categoria        | Puntuacion |
|------------------|------------|
| Arquitectura     | 8.5/10     |
| CalidadCodigo    | 8.0/10     |
| Buenas Practicas | 8.5/10     |
| Seguridad        | 8.5/10     |
| Documentacion    | 8.0/10     |
| Dependencias     | 8.0/10     |
| Deuda Tecnica    | 8.5/10     |
| GLOBAL           | 8.3/10     |

### Estadisticas delCodigo

- \* Lineas codigo (src): ~5,600
- \* Lineas tests: ~2,500
- \* Archivos Python: 40+
- \* Tests automatizados: 225

**Estado: ENTERPRISE-READY**