# Estructura del Proyecto Python - Exportador IBM i → Excel con FastAPI

Este documento describe la estructura recomendada para un proyecto Python que implementa una API REST con FastAPI para exportar datos desde IBM i (AS/400) a archivos Excel usando JDBC (jaydebeapi). Incluye seguridad con API Key, validaciones y separación modular por responsabilidades.

## 1. Estructura de Carpetas y Archivos

ibmi-export-api/  
├─ app/  
│ ├─ \_\_init\_\_.py  
│ ├─ main.py # Punto de entrada FastAPI  
│ ├─ config.py # Configuración (env vars, rutas, host IBM i)  
│ ├─ security.py # Dependencia de API Key (X-API-Key)  
│ ├─ schemas.py # Modelos Pydantic  
│ ├─ exporter.py # Lógica de exportación a Excel  
│ └─ services/  
│ ├─ jdbc.py # Conexión JDBC (jaydebeapi)  
│ └─ io\_utils.py # Validaciones y utilidades de I/O  
├─ jt400/ # (Opcional) carpeta para jt400.jar  
│ └─ jt400.jar  
├─ output/ # Carpeta de salida de .xlsx  
├─ .env.example # Variables de entorno de ejemplo  
├─ requirements.txt  
├─ README.md  
├─ run\_dev.bat # Script ejecución en Windows  
└─ run\_dev.sh # Script ejecución en Linux/macOS

## 2. requirements.txt

fastapi  
uvicorn  
pydantic>=2  
jaydebeapi  
jpype1  
pandas  
openpyxl  
python-dotenv

## 3. .env.example

EXPORT\_API\_KEY=MiSecreto\_2025  
IBMI\_HOST=10.246.17.67  
IBMI\_USER=ALDNOVACOM  
IBMI\_PASSWORD=ALDNOVACOM  
JAR\_PATH=C:\Users\distu\Documents\IBM\jt400.jar  
OUTPUT\_DIR=C:\Users\distu\OneDrive\Documentos\bel\_excel\conexion  
DEFAULT\_SCHEMA=DACCYFILES

## 4. Archivos Principales

main.py - Punto de entrada de la API:

from fastapi import FastAPI, Depends, HTTPException  
from .security import require\_api\_key  
from .schemas import ExportRequest, ExportResponse  
from .exporter import export\_to\_excel  
  
app = FastAPI(title='Exportador IBM i → Excel')  
  
@app.post('/export', response\_model=ExportResponse)  
def exportar(req: ExportRequest, \_: bool = Depends(require\_api\_key)):  
 try:  
 return export\_to\_excel(req.tabla, req.archivo)  
 except Exception as e:  
 raise HTTPException(status\_code=400, detail=str(e))

config.py - Configuración del proyecto con variables de entorno.

security.py - Función require\_api\_key que valida el header X-API-Key.

schemas.py - Modelos Pydantic para request/response.

exporter.py - Lógica principal de exportación a Excel.

services/jdbc.py - Conexión JDBC a IBM i.

services/io\_utils.py - Validaciones de nombres y utilidades de I/O.

## 5. Scripts de Ejecución

run\_dev.bat - Script para desarrollo en Windows con uvicorn.

run\_dev.sh - Script para desarrollo en Linux/macOS con uvicorn.

## 6. Notas

- La carpeta jt400/ es opcional y puede contener jt400.jar.  
- La carpeta output/ almacenará los archivos Excel exportados.  
- Las variables de entorno deben configurarse en un archivo .env en el directorio raíz.  
- requirements.txt contiene las dependencias necesarias.  
- El API se protege con X-API-Key definida en EXPORT\_API\_KEY.