

Campistas. Alejandro G. Jeison A. Sebastian D.
Andres R.

Reporte Técnico: Validador de Facturas DIAN

Descripción de la Solución

El **Validador de Facturas DIAN** es una aplicación web que automatiza el proceso de validación de facturas comerciales de importación según los estándares definidos en la Cartilla CT-COA-0124 de la DIAN. El sistema implementa un enfoque híbrido que combina validaciones determinísticas basadas en reglas aduanales con análisis inteligente contextual, permitiendo detectar tanto errores estructurales como inconsistencias sutiles en documentos de comercio internacional.

Herramientas y Tecnologías Utilizadas

Backend

- **FastAPI** (Framework web asincrónico de Python)
- **Pydantic** (Validación de datos)
- **Python 3.10+** (Lenguaje de programación)

Frontend

- **HTML + CSS + JavaScript**: Arquitectura simple sin dependencias externas de frameworks frontend. Comunica con backend mediante API REST usando fetch().
- **Tailwind CSS** (Framework de estilos): Generación de interfaz responsive y moderna mediante utility-first CSS. Incluye configuración personalizada de colores alineados con identidad visual.
- **Font Awesome**: Conjunto de iconografía SVG para mejorar experiencia visual.

Infraestructura

- **Uvicorn** (Servidor ASGI): Servidor web de producción para ejecutar FastAPI con soporte para aplicaciones asincrónicas.
- **CORS Middleware**: Permite que el frontend acceda al backend desde diferentes orígenes, facilitando desarrollo y despliegue descentralizado.

Reglas de Validación Implementadas

Validaciones Determinísticas (11 reglas)

1. **Tipo de Documento** (DIAN_001): Rechaza facturas pro forma; solo acepta facturas comerciales definitivas.
2. **Número de Factura** (DIAN_002): Verifica presencia y no-vacuidad del número factura único.
3. **Datos del Vendedor** (DIAN_003, DIAN_004): Valida nombre completo (≥ 3 caracteres) y dirección completa (≥ 10 caracteres) del proveedor.
4. **Datos del Comprador** (DIAN_005): Valida presencia de nombre completo del comprador.

5. **Fecha de Expedición** (DIAN_006, DIAN_007): Verifica formato parseable, rechaza fechas futuras, alerta sobre facturas con más de un año.

6. **Descripciones de Mercancía** (DIAN_008): Requiere descripciones de al menos 10 caracteres para evitar genericidad; integra análisis contextual con IA.

7. **Coherencia Numérica** (DIAN_009, DIAN_010): Valida que cantidad \times precio_unitario = precio_total (con tolerancia del 1% o \$1 mínimo); verifica suma de items = total_factura.

8. **Moneda** (DIAN_011): Verifica presencia de código ISO de moneda; alerta sobre monedas no comunes (USD, EUR, COP, CNY, GBP, JPY).

9. **Incoterms** (DIAN_012): Valida que Incoterm pertenezca a lista oficial (Incoterms 2020: EXW, FCA, FOB, CIF, DDP, etc.).

10. **Puertos y Origen** (DIAN_013, DIAN_014): Verifica presencia de puertos de carga/descarga y país de origen según normativa aduanal.

Validaciones Inteligentes (con API externa)

- **Análisis de Descripciones**: Utiliza análisis contextual para detectar si descripciones contienen marca, modelo y características técnicas suficientes.
- **Coherencia General**: Verifica consistencia entre campos relacionados (p. ej., moneda \leftrightarrow país proveedor, Incoterm \leftrightarrow puertos).

Supuestos Realizados

1. **Formato JSON estandarizado**: Las facturas deben llegar en estructura {"Fields": [...], "Table": [...] } con campos específicos (Supplier, Customer, InvoiceNumber, etc.). Las variaciones de formato requieren preprocesamiento.

2. **Valores numéricos como strings**: Cantidad y precios vienen como strings con separadores de miles (p. ej., "26,308.62"). Se convierte a float eliminando comas.

3. **Fechas en múltiples formatos**: El sistema intenta parsear fechas en formatos ISO, MM/DD/YYYY, DD-MMM-YYYY y DD/MM/YYYY. Otros formatos no se reconocen.

4. **Sin autenticación**: Versión actual asume acceso público; no implementa autenticación de usuario ni control de roles.

Limitaciones Actuales

1. **Ánalisis de IA limitado a 7 items**: Facturas con > 7 líneas solo tienen análisis IA en los primeros 7 items para controlar costos de API.

2. **Descripción de mercancía estática**: Validaciones de descripción se basan en longitud mínima y

palabras clave. No distingue entre "componentes electrónicos" (válido) vs "productos" (genérico) sin análisis contextual.

3. **Sin persistencia de datos:** No hay base de datos; cada sesión es independiente. Histórico de validaciones no se guarda.

4. **Cambio de moneda manual:** No convierte valores entre monedas. Si se requiere validar coherencia de precios en diferentes monedas, requiere pre-procesamiento.

Posibles Mejoras Futuras

Corto Plazo

- **Persistencia:** Agregar base de datos PostgreSQL para guardar histórico de validaciones y reportes analíticos.
- **Autenticación:** Implementar JWT y roles de usuario (admin, usuario, auditor) con control de acceso granular.
- **Batch validation:** Endpoint para validar múltiples facturas simultáneamente con generación de reportes en Excel.
- **Exportación de resultados:** Permitir descargar reportes en PDF con detalles de validación.

Mediano Plazo

- **Machine Learning:** Entrenador modelo sobre histórico de facturas para detectar patrones anómalos sin reglas explícitas.
- **OCR para PDF:** Extraer datos directamente de PDF de facturas en lugar de requerir JSON pre-procesado.
- **Integración DIAN:** Conexión directa con APIs de DIAN para validar NIT de proveedores en tiempo real.
 - **Soporte multi-idioma:** Validaciones contextuales en otros idiomas de importadores internacionales.

Largo Plazo

- **API de terceros:** Exponer validador como servicio SaaS para integradores y plataformas logísticas.