7ª EDICIÓN

DATATÓN 2025

ANTICORRUPCIÓN

SISTEMA NACIONAL ANTICORRUPCION

SECRETARÍA EJECUTIVA SNA

Índice de Riesgo por Inconsistencias Patrimoniales (IRIP)

Equipo - Codifikados

- ZURISADAI BAUTISTA CASTRO
- MARIANA VIANEY CASTRO MENDEZ
- DAVID ISAI BALAM RODRIGUEZ
- EMIR EDUARDO UC POOT

1. Problema.

La Plataforma Digital Nacional concentra miles de declaraciones patrimoniales, pero gran parte de ellas presentan altos niveles de datos faltantes, valores incongruentes, patrimonios ilógicos, o evoluciones patrimoniales difíciles de rastrear.
Estas inconsistencias dificultan la detección de posibles casos de ocultamiento de bienes, enriquecimiento irregular o manipulación de declaraciones, riesgos vinculados a faltas administrativas (LGRA, artículos 52–61), etc. El propósito de este reto es identificar **cambios incrementales fuera del rango de percepción de ingresos o patrimonio** para apoyar a la investigación anticorrupción.

Sin embargo, la información disponible presenta desafíos como:

- Declaraciones patrimoniales con **alto nivel de datos faltantes** (hasta 95% en variables clave).
- El volumen del dataset es significativo (**16,047 archivos JSON**), lo que dificulta un análisis manual o tradicional.
- Estructura de ingresos compleja (sueldo, profesional, industrial, financiero, otros).
- Inconsistencias entre el total declarado y la suma de los ingresos.
- Declaraciones sin patrimonio declarado, incluso en servidores con altos ingresos.

2. Objetivo General.

Diseñar y desarrollar un **Índice de Riesgo por Inconsistencias Patrimoniales** que permita detectar anomalías e indicios de riesgo de declaraciones con señales de alerta relacionadas con patrimonios e ingresos **atípicos**, **inconsistentes** o **potencialmente riesgosos**, usando los datos públicos del Sistema 1 de la PDN, combinando la minería de datos, reglas anticorrupción y análisis estadístico.

3. Objetivos Específicos.

1. **Integrar, depurar y estandarizar** los datos JSON del Sistema 1 de la PDN, considerando limitaciones computacionales mediante muestreo del 10%.

2. **Construir variables derivadas (features)** orientadas a auditoría: proporciones de ingresos, indicadores de patrimonio, conteo de nulos, valores críticos y banderas rojas.

3. **Aplicar métodos de detección de anomalías (Isolation Forest)** para identificar patrones fuera de la distribución normal del dataset.

4. **Diseñar y calcular un índice de riesgo anticorrupción (IRIP)** basado en:

- Congruencia económica
- Completitud
- Coherencia interna
- Reglas anticorrupción

5. **Clasificar a los servidores públicos:** en riesgo bajo, medio y alto, priorizando casos relevantes para auditoría.

6. **Generar visualizaciones y un dataset final:** apto para tablero, análisis exploratorio o sistemas institucionales futuros.

4. Metodología.

4.1. Recolección y Muestreo

- Carga de 16,047 declaraciones JSON.
- Por restricciones de RAM, se trabajará inicialmente con una muestra aleatoria del 10%, equivalente a ~695,813 registros (dato verificado en metadata).
- El pipeline es replicable al **100%** para obtener resultados finales.

Para cada archivo de la muestra:

- Se carga el contenido con `json.load`.
- Cada archivo contiene una lista de declaraciones; se itera sobre cada declaración para extraer únicamente los campos relevantes.

- Esto permite construir una lista de filas con las variables de interés para cada persona servidora pública.

4.2. Limpieza y Preparación

1. **Normalización de tipos y columnas numéricas**

- Se definió una lista de columnas numéricas potenciales (num_cols) que incluye ingresos, totales patrimoniales y, cuando están presentes, total_ingresos, patrimonio_bruto y patrimonio_neto.
- Solo se conservaron en num_cols las columnas que realmente existen en data.

2. **Limpieza de la columna anio y creación de sueldo**

- anio se convirtió a numérico con manejo de errores (errors='coerce'), asignando NaN a valores no válidos.
- Se filtraron años fuera del rango, asignándoles NaN.
- Se creó la columna sueldo como copia de ingreso_neto cuando esta columna existe.

3. **Manejo de valores negativos en variables numéricas**

- Para cada columna en num_cols: se generó una bandera flag_neg_<col> que vale 1 si el valor es negativo, 0 en caso contrario.
- Los valores negativos se reemplazaron por NaN para evitar distorsionar las métricas.
- A partir de estas banderas, se creó flag_valor_negativo como un indicador global (1 si hay al menos un valor negativo en el registro, 0 en caso contrario).

4. **Cálculo de porcentaje de nulos en patrimonio**

- Se definió un subconjunto de columnas patrimoniales: patrimonio_bruto, patrimonio_neto, inmuebles_total, vehiculos_total, muebles_total, adeudos_total (cuando existen).
- Se calculó **pct_nulos_patrimonio** como la proporción de valores nulos en estas columnas para cada registro.

4.3. Reglas de riesgo (banderas de inconsistencia)

Sobre el `DataFrame data` se definieron varias banderas lógicas para capturar patrones de riesgo:

1. **Ingreso alto + patrimonio en cero**

- Se calculó `total ingresos` (solo donde `total_ingresos > 0`).
- La bandera `flag_ingreso_alto_patrimonio_cero` se activa (1) cuando:
- `total_ingresos` ≥ umbral de ingreso alto, **y** `patrimonio_bruto` es 0 o nulo (se considera 0 tras `fillna(0)`).

2. **Declaración casi vacía en patrimonio**

- Se definió **`flag_declaracion_casi_vacia`** como 1 cuando `pct_nulos_patrimonio` ≥ 0.9, es decir, cuando **más del 90%** del patrimonio está vacío o no declarado.

3. **Declaración con todo en cero (ingresos + patrimonio)**

- Se construyó una lista de columnas de ingreso (`ing_cols`) y se usaron las columnas patrimoniales (`pat_cols`).
- La bandera `flag_todo_cero` se activa cuando:
- Todos los ingresos (`ing_cols`) son 0 o nulos, **y**
- Todos los componentes de patrimonio (`pat_cols`) son 0 o nulos.

4. **Cambios bruscos entre declaraciones del mismo ID**

- Se ordena la información por `id` y `anio`.
- Para las columnas `total_ingresos` y `patrimonio_bruto` (si existen), se calculó el cambio porcentual año a año por persona (`groupby('id').pct_change()`), generando columnas del tipo `pct_cambio_total_ingresos`.
- Se crearon banderas específicas `flag_brusco_<col>` que se activan cuando el cambio absoluto es mayor al 200% (`abs(pct_cambio) > 2.0`).
- La bandera global `flag_cambio_brusco` se activa cuando al menos una de estas banderas específicas es 1.

4.4. Transformaciones logarítmicas y selección de variables para el modelo

Para estabilizar escalas y reducir el impacto de valores extremos, se aplicaron transformaciones logarítmicas seguras (np.log1p) sobre las columnas numéricas patrimoniales y de ingresos totales:

- total_ingresos_log
- patrimonio_bruto_log
- patrimonio_neto_log
- inmuebles_total_log
- vehiculos_total_log
- muebles_total_log
- adeudos_total_log

Posteriormente se definió el conjunto de variables de entrada del modelo:

1. Variables transformadas (log_cols_modelo): todos los campos *_log.
2. Variables de banderas e indicadores (flag_cols_modelo):

- flag_ingreso_alto_patrimonio_cero
- flag_declaracion_casi_vacia
- flag_todo_cero
- flag_valor_negativo
- flag_cambio_brusco
- pct_nulos_patrimonio

4.5. Modelo de riesgo: Isolation Forest

Para la detección de anomalías se construyó un pipeline con:

- SimpleImputer
- StandardScaler
- IsolationForest (n_estimators=200, contamination=0.05)

El modelo asignó un score de anomalía a cada registro, posteriormente normalizado a escala 0–100 (riesgo_score).
Se clasificaron los niveles de riesgo en:

- Bajo (<50)
- Medio (50–79)
- Alto (≥80)

4.6. Generación de salidas para dashboard y metadatos

1. Exportación de resultados a CSV

Se guardó un CSV con todas las columnas del DataFrame (resultados_anticorrupcion_v2.csv), y posteriormente se generó una versión orientada al dashboard (resultados_anticorrupcion.csv), incluyendo:

- Identificadores (id, nombre, apellidos, institución, cargo, año).
- Variables clave (ingreso_cargo, otros_ingresos, total_ingresos, patrimonio_bruto).
- Proporción de otros ingresos sobre el total (prop_otros_ingresos).
- Puntajes y niveles de riesgo (riesgo_score, riesgo_nivel, riesgo_modelo, anomaly_iforest).
- Se incluyó una estructura de reglas anticorrupción (`R1`…`R10`) y un puntaje adicional (`score_reglas`) para versiones futuras.
- Estructura para futuras reglas R1–R10 (score_reglas, R1_..., R2_..., ..., R10_...), inicializadas en esta versión en 0 o NaN para evitar errores en el dashboard.

Se generó un CSV final con variables clave, puntajes y niveles de riesgo, listo para dashboard y se creó un archivo **metadata_riesgo.json** que documenta:

- fecha del análisis
- porcentaje de muestra
- columnas usadas por el modelo
- distribución final de los niveles de riesgo

5. Resultados Principales.

Los resultados obtenidos permiten identificar patrones de riesgo patrimonial mediante la combinación de análisis estadístico, banderas lógicas y detección de anomalías. Con base en la muestra analizada (~10% de las declaraciones del Sistema 1), se obtuvieron los siguientes hallazgos:

1. **Completitud de la información**

- Entre 80% y 95% de los registros presentan información patrimonial incompleta o totalmente nula.
- La variable pct_nulos_patrimonio confirma que el 90% de las declaraciones carecen de datos suficientes para evaluar patrimonio de manera tradicional.

2. **Señales de alerta identificadas**

   El sistema detectó patrones relevantes como:

- Ingreso alto sin patrimonio declarado (`flag_ingreso_alto_patrimonio_cero`).
- Declaraciones casi vacías (`flag_declaracion_casi_vacia`).
- Declaraciones con todos los valores en cero (`flag_todo_cero`).
- Cambios abruptos entre declaraciones consecutivas (`flag_cambio_brusco`).

3. **Resultados del modelo**

El modelo detectó el 5% de valores atípicos en la muestra, coherente con el nivel de contaminación definido.

Los registros fueron agrupados por nivel de riesgo:

Nivel de riesgo	Casos	Porcentaje
Bajo	654,066	94%
Medio	39,628	5.7%
Alto	2,119	0.3%

6. Conclusiones

1. **La detección de anomalías es viable incluso con datos altamente incompletos.**
   El modelo logra identificar patrones inusuales sin depender de información patrimonial completa.

2. **Las banderas lógicas permiten explicar de forma clara dónde están las inconsistencias.**
   Esto es esencial para auditores y analistas que requieren interpretabilidad, no solo señales numéricas.

3. **El Índice de Riesgo (basado en Isolation Forest) es una herramienta efectiva de priorización.**
   No acusa corrupción, pero sí identifica casos que deben revisarse con mayor atención.

4. **El sistema es escalable y reproducible.** Puede aplicarse al dataset completo del Sistema 1 y adaptarse a otros sistemas de la PDN.

5. **Los hallazgos confirman una problemática estructural: la mayoría de las declaraciones se presentan incompletas**. Dificulta la auditoría tradicional y resalta la necesidad de herramientas como esta.

7. Limitaciones

- **La alta proporción de nulos** en patrimonio limita la precisión del análisis.

- **El modelo depende únicamente de la información declarada**, sin cruces con otros sistemas (contrataciones, intereses, empresas, etc.).

- **El 10% de muestra** —aunque representativo— no sustituye el análisis total del dataset completo.

- **Los resultados son probabilísticos**, no deterministas; el modelo identifica riesgo, no culpabilidad.

- **Las reglas anticorrupción aún no están integradas completamente** en la versión actual del modelo (preparadas pero sin afectar el score final).

8. Trabajos Futuros

- Integrar información de contrataciones (Sistema 6).
- Asociar empresas vinculadas a familiares (declaración de intereses).
- Usar modelos de series temporales para evolución patrimonial por año.
- Crear dashboard en Streamlit para SESNA.