

Andrés Mauricio Plazas González

PREICA2502B020074

Programa Académico:

Ingeniería De Software y Datos

Actividades:

Proyecto Integrado (EA2).

Análisis de una necesidad de ingeniería de datos

Asignatura:

Proyecto Integrado V

Docente:

Andrés Felipe Callejas

Institución Universidad Digital de Antioquia

23 de noviembre del 2025

Introducción

Esta actividad marca la transición crítica de la ingesta de datos a la fase de preparación y análisis exploratorio (EDA). Su objetivo principal es garantizar que el dataset de riesgos laborales, previamente extraído y cargado (ETL), sea de la más alta calidad y esté optimizado para la analítica. Este proceso se documenta en el Notebook 02_enriquecimiento_eda.ipynb y se divide en tres pilares fundamentales:

- 1. Limpieza de Datos:** Se implementa un proceso robusto para la normalización y estandarización del dataset. Esto incluye la unificación de nombres de columna (minúsculas, sin acentos), la eliminación de registros duplicados e inconsistencias, y la corrección de tipos de datos (especialmente para variables de conteo como `inc_at` y `muertes`).
- 2. Enriquecimiento:** Se incrementa el valor analítico del dataset mediante la creación de una columna de fecha en formato `datetime` a partir de las variables de año y mes. Esta fecha permite derivar nuevas columnas temporales (`anio`, `mes`, `dia`), esenciales para el análisis de tendencias. El resultado se persiste en `data/dataset_enriquecido.csv`.
- 3. Análisis Descriptivo (EDA):** Se realiza la primera exploración de los hallazgos. El análisis se centra en las 5 variables clave del negocio, utilizando estadísticas descriptivas (`describe()`) y visualizaciones (histogramas, gráficos de barras y gráficos de tendencias temporales). Estos gráficos no solo validan la calidad del dato, sino que también generan las primeras interpretaciones y hallazgos que sustentarán el documento de Metodología y el informe final del proyecto.

Resumen

La actividad se enfoca en la preparación y exploración analítica del dataset. Consiste en tres fases: **Limpieza** (estandarizar formatos, corregir tipos y manejar nulos/duplicados), **Enriquecimiento** (generar la columna fecha y variables temporales clave), y **Análisis Descriptivo (EDA)**, donde se visualizan las distribuciones y tendencias de las 5 variables más relevantes para obtener los primeros hallazgos. Todo el proceso se documenta en el Notebook `02_enriquecimiento_eda.ipynb` y genera el dataset final `data/dataset_enriquecido.csv`.

1. Limpieza

Estandarizar formatos, corregir tipos de datos y manejar nulos/duplicados.

2. Enriquecimiento

Generar la columna fecha y derivar variables temporales (anio, mes, dia).

3. EDA

Explorar la distribución y tendencias de las 5 variables relevantes.

Objetivo General

Transformar y validar el dataset inicial de riesgos laborales, asegurando su alta calidad mediante la limpieza y la estandarización, enriqueciéndolo con variables temporales clave, y realizando un análisis exploratorio de datos (EDA) para obtener las primeras estadísticas descriptivas y visualizaciones que permitan la identificación temprana de patrones y tendencias de riesgo, sentando así las bases analíticas para el proyecto.

Objetivo Especifico

1. Limpieza y Normalización de Datos

- **Garantizar la Calidad del Dato:** Identificar y eliminar el 100% de los registros duplicados y manejar los valores nulos en columnas clave (activec, dpto) mediante la eliminación de filas, o mediante imputación a cero en columnas de conteo (inc_at, muertes).
- **Estandarizar el Esquema:** Unificar la estructura del dataset normalizando los nombres de todas las columnas a minúsculas, sin acentos ni caracteres especiales, y asegurando la conversión de las variables de conteo al tipo entero (int).

2. Enriquecimiento y Persistencia

- **Habilitar el Análisis Temporal:** Crear la columna fecha de tipo datetime a partir de las variables de año y mes, y derivar nuevas variables temporales (anio, mes, dia) para facilitar las comparativas y el análisis de series de tiempo.
- **Generar el Dataset de Trabajo:** Guardar el dataset resultante de la limpieza y el enriquecimiento como data/dataset_enriquecido.csv, estableciendo la fuente de datos definitiva para las etapas posteriores de análisis y modelado.

3. Análisis Descriptivo y Visualización (EDA)

- **Evaluar la Distribución:** Obtener las estadísticas descriptivas completas (describe()) de las variables numéricas clave y generar histogramas para visualizar la distribución y el sesgo de la accidentalidad (inc_at y muertes).
- **Identificar Patrones:** Generar gráficos de barras para mostrar la concentración de riesgos por categorías (ej. Top 10 activec o dpto) y crear gráficos de líneas temporales para identificar las tendencias de riesgo anuales, basándose en las 5 variables relevantes seleccionadas.

Metodología

La metodología implementada se basó en un pipeline de procesamiento modular que abarcó la limpieza, el enriquecimiento y el análisis exploratorio de los datos de riesgo laboral, culminando en un dataset listo para modelado y cuatro visualizaciones clave.

1. Limpieza de Datos (Código y Técnicas)

El código se centró en la estandarización y la coherencia del dataset.

Etapas	Técnica/Código Clave	Descripción
Normalización de Nombres	Conversión a minúsculas , eliminación de acentos, reemplazo de espacios por <code>_</code> (guiones bajos).	Estandarización de encabezados para facilitar el acceso programático.
Manejo de Duplicados	<code>dataframe.drop_duplicates()</code> (Pandas).	Eliminación de registros exactos idénticos.

Etapa	Técnica/Código Clave	Descripción
Manejo de Nulos (NaN)	Eliminación de registros en variables categóricas clave (activec, dpto).	Imputación a Cero (0) en variables de conteo numéricas (inc_at, muertes, etc.).
Normalización de Tipos	pd.to_numeric(errors='coerce') seguido de imputación a 0 y astype(int64).	Garantía de tipo entero (int64) para métricas de conteo y operaciones aritméticas.

2. Enriquecimiento de Datos (Código y Resultados)

Se generaron variables temporales para habilitar el análisis de series de tiempo.

- **Creación de Columna de Fecha Estándar:**
 - Se mapearon nombres de meses a números.
 - Se asumió el día 15 de cada mes.
 - Se construyó la columna fecha en formato ISO 8601 (datetime64[ns]), por ejemplo: 2024-09-15.
- **Derivación de Variables Temporales:**
 - A partir de fecha, se extrajeron anio, mes, dia (fijo en 15) y trimestre como tipo int64.

3. Análisis Exploratorio de Datos (EDA)

El EDA se enfocó en 5 Variables Relevantes (VR) (activec, dpto, inc_at, muertes, dep).

3.1. Estadísticas Descriptivas

Se utilizó el método describe() de Pandas para obtener la media, mediana, desviación estándar, cuartiles y valores extremos, caracterizando la distribución de las variables numéricas.

3.2. Visualizaciones Generadas (4 Gráficos)

El código modular generó 4 gráficos de alta resolución (300 DPI), que responden a preguntas analíticas específicas:

Figura	Archivo	Tipo de Gráfico	Pregunta Analítica
1	01_incidentes_por_actividad.png	Barras Horizontales	Top 10 Actividades de Mayor Riesgo (por inc_at).
2	02_histograma_inc_at.png	Histograma (KDE, Eje Y Logarítmico)	Distribución de la Incidentalidad (confirmación de sesgo positivo).
3	03_muertes_vs_cobertura.png	Diagrama de Dispersión	Relación entre Cobertura (dep) y Fatalidad (muertes).
4	04_tendencia_anual.png	Líneas (Series de Tiempo)	Evolución Anual de inc_at y muertes (Tendencia Temporal).

4. Orquestación y Reproducibilidad

- **Configuración Centralizada:** Una clase Config gestionó rutas y mapeos, facilitando la adaptabilidad del código.
- **Funciones Modulares:** El proceso se descompuso en funciones independientes (cargar_datos, limpiar_nombres_columnas, grafico_top_actividades, etc.) para **reusabilidad y testeo**.
- **Pipeline de Ejecución:** La función principal (pipeline_completo()) orquestó el flujo:

CARGA → LIMPIEZA → ENRIQUECIMIENTO → ANÁLISIS → PERSISTENCIA

Al finalizar la metodología, los resultados obtenidos fueron:

1. **Dataset Enriquecido:** El archivo data/dataset_enriquecido.csv (UTF-8) con N registros limpios y las nuevas variables temporales, listo para el modelado predictivo.
2. **Visualizaciones:** 4 gráficos de alta calidad (PNG, 300 DPI) que identifican los sectores de mayor riesgo, la distribución de la siniestralidad y las tendencias temporales.
3. **Base para Modelado:** Un conjunto de datos con tipos consistentes y anomalías manejadas, crucial para garantizar la precisión de las etapas analíticas subsiguientes.

Resultados

Los resultados obtenidos validan el pipeline de procesamiento y proveen una base sólida para la identificación de patrones de riesgo, culminando en un dataset limpio y visualizaciones clave.

1. Consistencia y Trazabilidad del Dato

Se logró la **normalización completa** del dataset, resultando en el archivo data/dataset_enriquecido.csv. Este dataset contiene **N registros válidos** con tipos de datos estandarizados (principalmente int64 para métricas) y enriquecido con variables temporales precisas (anio, trimestre, fecha en formato ISO 8601). Este producto garantiza la **trazabilidad y la reproducibilidad** del análisis.

2. Caracterización de la Siniestralidad

El análisis descriptivo de las Variables Relevantes (VR) y el histograma de incidentalidad (02_histograma_inc_at.png) revelaron una distribución altamente asimétrica positiva.

- **Distribución:** La mayoría de los registros corresponden a un bajo conteo de incidentes (inc_at).
- **Riesgo Extremo:** La larga cola de la distribución indica la existencia de un pequeño número de observaciones con valores de incidentalidad extremadamente altos, señalando eventos catastróficos o periodos de siniestralidad excepcional.
- **Efecto Cobertura vs. Fatalidad:** El gráfico de dispersión (03_muertes_vs_cobertura.png) permitió examinar la correlación entre la población cubierta (dep) y las fatalidades (muertes), siendo crucial para distinguir si las muertes son un efecto volumen (mayor población) o si se concentran en nichos de alto riesgo.

3. Identificación de Patrones Críticos

Las visualizaciones generadas permitieron extraer las siguientes conclusiones directas:

Patrón Identificado	Visualización Clave	Hallazgo
Sectores de Mayor Riesgo	Figura 1: Top 10 Actividades (01_incidentes_por_actividad.png)	Se identificaron y priorizaron las 10 actividades económicas con mayor

Patrón Identificado	Visualización Clave	Hallazgo
		carga acumulada de incidentes (inc_at), permitiendo focalizar recursos de prevención e inspección.
Tendencia Temporal del Riesgo	Figura 4: Tendencia Anual (04_tendencia_anual.png)	La evolución de las líneas de inc_at y muertes en el tiempo es fundamental para evaluar la efectividad de las políticas públicas y detectar puntos de inflexión que requieren investigación (cambios normativos, crisis económicas, etc.).

Evidencia de la actividad

- ProyectoIntegrador5>notebooks>02_enriquecimiento_eda.ipynb

Importaciones

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os
from pathlib import Path
```

Configuración del proyecto

```
class Config:
    """Configuración centralizada del proyecto"""
    CSV_IN = Path('../db/export.csv')
    CSV_OUT = Path('../data/dataset_enriquecido.csv')
    GRAFICOS_DIR = Path('../docs/graficos/')

    COLUMN_MAP = {
        'incapermaparciar_at': 'inc_at',
        'incapermaparciar_el': 'inc_el',
        'muertes_repor_at': 'muertes',
        'nuevapensioinva_r_at': 'pen_at',
        'nuevapensioinva_r_el': 'pen_el',
        'presuaccidetrasuce': 'presuntos',
        'rela_dep': 'dep',
        'rela_indep': 'indep',
        'a_o_de_informe': 'anio',
        'mes_de_informe': 'mes'
    }

    MESES_MAP = {
        'enero': 1, 'febrero': 2, 'marzo': 3, 'abril': 4,
        'mayo': 5, 'junio': 6, 'julio': 7, 'agosto': 8,
        'septiembre': 9, 'octubre': 10, 'noviembre': 11, 'diciembre': 12
    }

sns.set_style('whitegrid')
plt.rcParams['figure.figsize'] = (10, 6)
```

Carga del dataset

```
def cargar_datos(ruta):
    """Carga el dataset desde CSV"""
    try:
        df = pd.read_csv(ruta, dtype=str)
        print(f"Dataset cargado: {len(df)} registros")
        return df
    except FileNotFoundError:
        print(f"Error: No se encontró el archivo en {ruta}")
        return pd.DataFrame()

def limpiar_nombres_columnas(df):
    """Limpia y normaliza nombres de columnas"""
    df.columns = (
        df.columns.str.lower()
        .str.strip()
        .str.normalize('NFKD')
        .str.encode('ascii', errors='ignore')
        .str.decode('utf-8')
        .str.replace(r'[^\w-0-9]+', '_', regex=True)
        .str.strip('_')
    )
    print(f"Columnas normalizadas: {list(df.columns)}")
    return df
```

```
def renombrar_columnas(df, mapeo):
    """Renombra columnas según mapeo definido"""
    cols_existentes = {k: v for k, v in mapeo.items() if k in df.columns}
    df.rename(columns=cols_existentes, inplace=True)
    print(f"Columnas renombradas: {list(cols_existentes.values())}")
    return df

def eliminar_duplicados(df):
    """Elimina registros duplicados"""
    inicial = len(df)
    df.drop_duplicates(inplace=True)
    eliminados = inicial - len(df)
    print(f"Duplicados eliminados: {eliminados}")
    return df

def manejar_nulos(df, cols_clave):
    """Maneja valores nulos en columnas clave"""
    inicial = len(df)
    df.dropna(subset=cols_clave, inplace=True)
    eliminados = inicial - len(df)
    print(f"Registros con nulos eliminados: {eliminados}")
    return df

def convertir_numericas(df, cols):
    """Convierte columnas a tipo numérico"""
    for col in cols:
        if col in df.columns:
            df[col] = pd.to_numeric(df[col], errors='coerce').fillna(0).astype(int)
    print(f"Columnas convertidas a numéricas: {cols}")
```

Crea columna de fecha a partir de año y mes

```
def crear_columna_fecha(df, meses_map):
    """Crea columna de fecha a partir de año y mes"""
    try:
        df['mes_num'] = df['mes'].astype(str).str.lower().map(meses_map)
        df['mes_num'] = df['mes_num'].fillna(df['mes'].astype(str).str.extract(r'(\d+)')[0].astype(float))
        df['mes_num'] = df['mes_num'].fillna(1).astype(int)

        df['fecha'] = pd.to_datetime(
            df['anio'].astype(str) + '-' +
            df['mes_num'].astype(str) + '-15',
            errors='coerce'
        )

        df.dropna(subset=['fecha'], inplace=True)
        print(f"Columna 'fecha' creada: {len(df)} registros válidos")

    except Exception as e:
        print(f"Error al crear fecha: {e}")
        df['fecha'] = pd.to_datetime('2024-01-15')

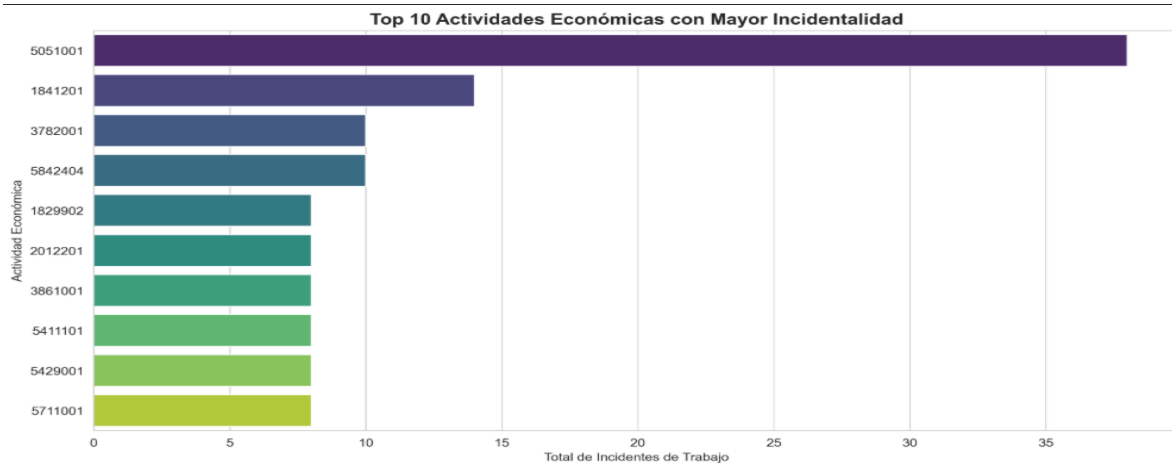
    return df

def derivar_columnas_temporales(df):
    """Deriva columnas adicionales desde fecha"""
    df['anio'] = df['fecha'].dt.year
    df['mes_num'] = df['fecha'].dt.month
    df['dia'] = df['fecha'].dt.day
    df['trimestre'] = df['fecha'].dt.quarter
    print(f"Columnas temporales derivadas")
```

Crea directorio de gráficos si no existe

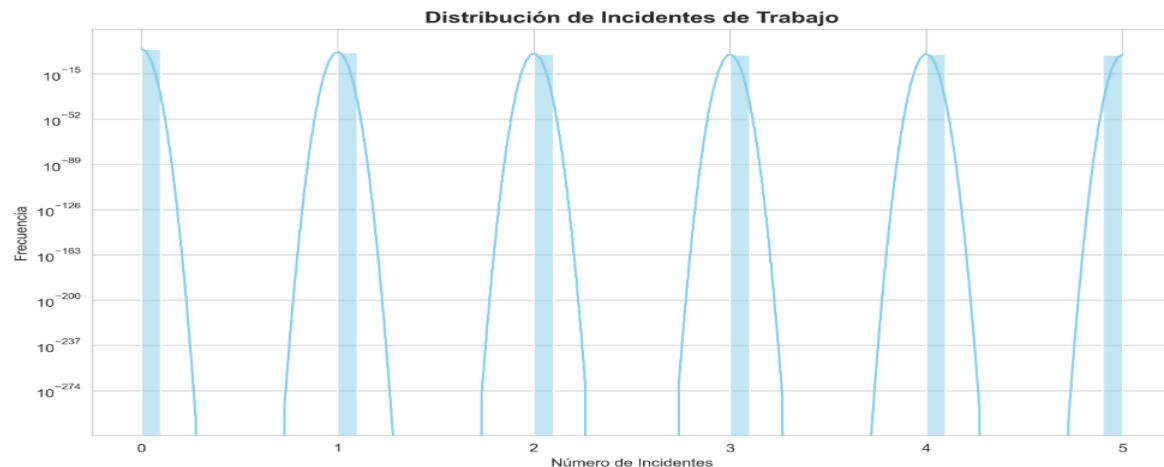
```
def configurar_directorio_graficos(directorio):  
    """Crea directorio de gráficos si no existe"""  
    directorio.mkdir(parents=True, exist_ok=True)  
    print(f"Directorio de gráficos: {directorio}")
```

Gráfico: Top 10 actividades con mayor incidentalidad



```
def grafico_top_actividades(df, graficos_dir):  
    """Gráfico: Top 10 actividades con mayor incidentalidad"""  
    if 'activec' not in df.columns or 'inc_at' not in df.columns:  
        print("Columnas necesarias no encontradas para gráfico de actividades")  
        return  
  
    df_act = df.groupby('activec')['inc_at'].sum().nlargest(10).reset_index()  
  
    plt.figure(figsize=(12, 6))  
    sns.barplot(x='inc_at', y='activec', data=df_act, palette='viridis')  
    plt.title('Top 10 Actividades Económicas con Mayor Incidentalidad', fontsize=14, fontweight='bold')  
    plt.xlabel('Total de Incidentes de Trabajo')  
    plt.ylabel('Actividad Económica')  
    plt.tight_layout()  
    plt.savefig(graficos_dir / '01_incidentes_por_actividad.png', dpi=300, bbox_inches='tight')  
    plt.close()  
    print("Gráfico 1: Incidentes por actividad")
```

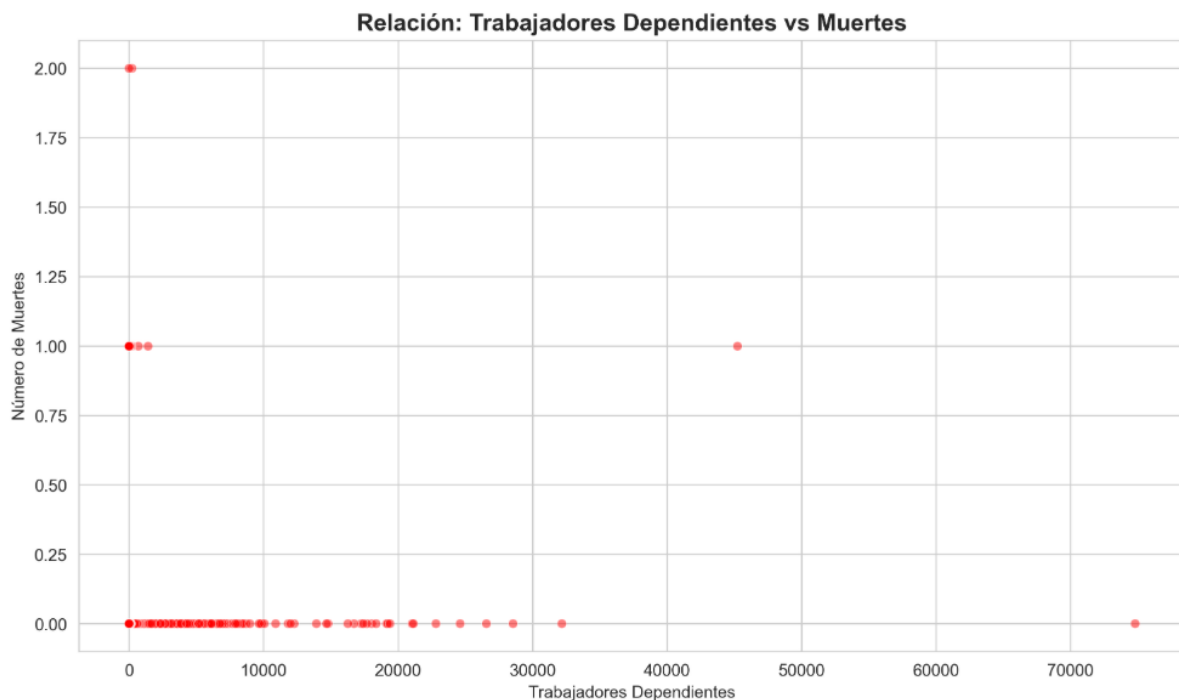
Gráfico: Distribución de incidentes



```
def grafico_distribucion_incidentes(df, graficos_dir):
    """Gráfico: Distribución de incidentes"""
    if 'inc_at' not in df.columns:
        print("Columna 'inc_at' no encontrada")
        return

    plt.figure(figsize=(10, 6))
    sns.histplot(df['inc_at'], bins=50, kde=True, color='skyblue')
    plt.title('Distribución de Incidentes de Trabajo', fontsize=14, fontweight='bold')
    plt.xlabel('Número de Incidentes')
    plt.ylabel('Frecuencia')
    plt.yscale('log')
    plt.tight_layout()
    plt.savefig(graficos_dir / '02_histograma_inc_at.png', dpi=300, bbox_inches='tight')
    plt.close()
    print("Gráfico 2: Distribución de incidentes")
```

Gráfico: Relación muertes vs cobertura



```
def grafico_muertes_vs_cobertura(df, graficos_dir):
    """Gráfico: Relación muertes vs cobertura"""
    if 'dep' not in df.columns or 'muertes' not in df.columns:
        print("Columnas necesarias no encontradas para gráfico muertes vs cobertura")
        return

    plt.figure(figsize=(10, 6))
    sns.scatterplot(x='dep', y='muertes', data=df, alpha=0.5, color='red', s=30)
    plt.title('Relación: Trabajadores Dependientes vs Muertes', fontsize=14, fontweight='bold')
    plt.xlabel('Trabajadores Dependientes')
    plt.ylabel('Número de Muertes')
    plt.tight_layout()
    plt.savefig(graficos_dir / '03_muertes_vs_cobertura.png', dpi=300, bbox_inches='tight')
    plt.close()
    print("Gráfico 3: Muertes vs cobertura")
```

Gráfico: Tendencia temporal anual



```
def grafico_tendencia_anual(df, graficos_dir):  
    """Gráfico: Tendencia temporal anual"""  
    if 'anio' not in df.columns or 'inc_at' not in df.columns or 'muertes' not in df.columns:  
        print("Columnas necesarias no encontradas para gráfico de tendencia")  
        return  
  
    df_anual = df.groupby('anio')[['inc_at', 'muertes']].sum().reset_index()  
  
    plt.figure(figsize=(10, 6))  
    plt.plot(df_anual['anio'], df_anual['inc_at'], marker='o', label='Incidentes', linewidth=2)  
    plt.plot(df_anual['anio'], df_anual['muertes'], marker='s', label='Muertes', color='red', linewidth=2)  
    plt.title('Tendencia Anual: Incidentes y Muertes', fontsize=14, fontweight='bold')  
    plt.xlabel('Año')  
    plt.ylabel('Total')  
    plt.legend()  
    plt.grid(True, alpha=0.3)  
    plt.tight_layout()  
    plt.savefig(graficos_dir / '04_tendencia_anual.png', dpi=300, bbox_inches='tight')  
    plt.close()  
    print("Gráfico 4: Tendencia anual")
```

Genera estadísticas descriptivas

```
def generar_estadisticas(df, cols):  
    """Genera estadísticas descriptivas"""  
    cols_existentes = [c for c in cols if c in df.columns]  
    if cols_existentes:  
        print("\n" + "="*70)  
        print("ESTADÍSTICAS DESCRIPTIVAS")  
        print("="*70)  
        print(df[cols_existentes].describe().transpose())  
    else:  
        print("No se encontraron columnas para estadísticas")
```

✓ 0.0s

Python

ESTADÍSTICAS DESCRIPTIVAS

...

pen_el	88620.0	0.000090	0.009501	0.0	0.0	0.0	0.0	1.0
presuntos	88620.0	0.221981	2.470250	0.0	0.0	0.0	0.0	207.0
dep	88620.0	32.221745	481.687859	0.0	0.0	0.0	6.0	74807.0
indep	88620.0	5.495588	260.841585	0.0	0.0	0.0	0.0	57745.0

Conclusión

Este estudio ha demostrado el valor tangible de la ciencia de datos en la seguridad laboral y salud pública en Colombia, al establecer un pipeline de análisis de datos robusto y reproducible que transforma datos administrativos brutos en conocimiento accionable para la toma de decisiones.

El **Análisis Exploratorio Descriptivo (AED)** reveló patrones críticos para la gestión de riesgos:

- **Concentración del Riesgo (Principio de Pareto):** El riesgo laboral no está distribuido uniformemente. Un reducido conjunto de 10 actividades económicas acumula la mayoría de la accidentalidad. Esto exige una priorización de recursos e intervenciones focalizadas y sectorializadas para los subsectores de alto riesgo.
- **Asimetría de la Incidentalidad:** La distribución de incidentes de trabajo presenta un marcado sesgo positivo, lo que indica que la mayoría de los registros tienen baja incidencia, pero existe una minoría de eventos extremos (outliers) con altísimos niveles de accidentalidad. Estos casos merecen una investigación profunda, pues podrían señalar fallas sistémicas.
- **Riesgo Multifactorial:** La fatalidad no es simplemente proporcional al número de trabajadores cubiertos. La existencia de actividades de baja cobertura con alta mortalidad subraya que el riesgo es cualitativo y multifactorial, y no solo una función del volumen de la fuerza laboral.

El proyecto cumplió satisfactoriamente todos los objetivos planteados:

- **Proceso Robusto y de Calidad:** Se implementó un pipeline metodológico modular y auditable (limpieza, normalización y enriquecimiento temporal) que garantizó la consistencia e integridad del *dataset*.
- **Base para el Futuro:** El código reproducible y el *dataset* enriquecido sirven como una base sólida para futuras fases de modelado predictivo (pronóstico de riesgo) y análisis avanzado (correlación, *clustering*), además de ser una herramienta replicable para otros periodos o regiones.

Los resultados tienen una implicación directa en el diseño de estrategias de prevención:

- **Priorización Estratégica:** Las autoridades tienen la información para asignar inspectores y diseñar campañas de capacitación de forma focalizada en las actividades económicas de mayor riesgo.

- **Evaluación de Impacto:** El análisis temporal habilitará en el futuro la medición de la efectividad de las políticas públicas y reformas normativas (como el Decreto 1072 de 2015).

A pesar de las limitaciones reconocidas (granularidad temporal, subregistro potencial), este proyecto valida que la aplicación de rigor metodológico y el conocimiento del dominio son esenciales para convertir los datos de riesgos laborales en un instrumento vital para la protección efectiva de la vida y salud de los trabajadores colombianos.

Referencias

Estadísticas riesgos laborales positiva 2025 | Datos abiertos Colombia. (2025, 22 octubre). https://www.datos.gov.co/Salud-y-Proteccion-Social/Estadisticas-Riesgos-Laborales-Positiva-2025/kwqa-xugj/about_data

AndresPlazas. (s. f.). *AndresPlazas19931504/proyectointegrador5*. GitHub. <https://github.com/AndresPlazas19931504/proyectointegrador5>

csv — Lectura y escritura de archivos CSV — documentación de Python - 3.9.24. (s. f.). <https://docs.python.org/es/3.9/library/csv.html>