Cuaderno reproducible

Escenarios de Riesgo ante el Fenómeno El Niño 2026–2027

Christian Euscátegui, Moises Santizo

Subdirección para el Conocimiento del Riesgo (SCR), UNGRD

CRC-006: ESCENARIOS DE RIESGO ANTE EL FENÓMENO EL NIÑO 2026–2027

Colombia es uno de los países de América Latina más afectados por el **Fenómeno El Niño (ENSO fase cálida)**, que altera de forma significativa los patrones de precipitación y temperatura en todo el territorio nacional. Sus efectos son diferenciados: en gran parte del país genera **déficit hídrico** que favorece incendios de la cobertura vegetal, desabastecimiento de agua y heladas; mientras que en algunas regiones puede intensificar las lluvias, aumentando el riesgo de inundaciones, movimientos en masa y avenidas torrenciales.

Este cuaderno aplica el **método del análogo histórico**: usa los registros oficiales de emergencias de la UNGRD durante el Niño 2023–2024 para construir escenarios de riesgo ante un eventual Fenómeno El Niño 2026–2027 de magnitud e intensidad comparable.

¿Qué aprenderás?

- Cargar y preprocesar datos de emergencias en formato Parquet con Pandas
- Clasificar eventos por **categoría climática** (condiciones lluviosas y secas)
- Calcular indicadores de impacto humanitario y de infraestructura
- Generar **mapas coropléticos interactivos** departamentales con Folium
- Crear rankings territoriales y análisis temporales con Altair
- Exportar resultados en CSV para uso en sistemas de información

1. Instalación de Dependencias

```
import sys

# Detección automática del entorno de ejecución
_ES_COLAB = 'google.colab' in sys.modules

paquetes = [
    'pandas>=2.0',
    'pyarrow>=12.0',
    'altair>=5.0',
    'folium>=0.14',
]

print('📦 Verificando dependencias...')
import subprocess
subprocess.run([sys.executable, '-m', 'pip', 'install', '-q'] + paquetes, check=True)
entorno = 'Google Colab' if _ES_COLAB else 'local / Binder'
print(f'✅ Entorno: {entorno}')
```

```
📦 Verificando dependencias...
✅ Entorno: local / Binder
```

2. Importación de Bibliotecas y Configuración

```
import pandas as pd
import json
import os
import warnings

import altair as alt
```

```
import folium
import matplotlib.pyplot as plt
from IPython.display import display

warnings.filterwarnings('ignore')
alt.data_transformers.disable_max_rows()

# Configuración global de matplotlib
plt.rcParams.update({
    'figure.dpi': 120,
    'font.family': 'sans-serif',
    'axes.spines.top': False,
    'axes.spines.right': False,
})

# Paleta institucional UNGRD
UNGRD_BLUE   = '#0154a5'
UNGRD_ACCENT = '#2563eb'
UNGRD_RED    = '#dc2626'

print('✅ Bibliotecas cargadas correctamente')
```

```
✅ Bibliotecas cargadas correctamente
```

3. Configuración de Rutas y Parámetros

```
# Rutas relativas al cuaderno (datos incluidos en el repositorio)
RUTA_DATOS      = os.path.join('data', 'datos_base_analisis.parquet')
RUTA_DEPTOS     = os.path.join('data', 'colombia_departamentos.json')
RUTA_MUNICIPIOS = os.path.join('data', 'colombia_municipios.json')

# Etiquetas de los trimestres del análogo en orden cronológico
ORDEN_TRIMESTRES = [
    'junio-agosto 2023',
    'septiembre-noviembre 2023',
    'diciembre 2023–febrero 2024',
    'marzo-mayo 2024',
]

TRIM_MAP = {
    'JJA_2023':      'junio-agosto 2023',
    'SON_2023':      'septiembre-noviembre 2023',
    'DEF_2023_2024': 'diciembre 2023–febrero 2024',
    'MAM_2024':      'marzo-mayo 2024',
}

# Variables de impacto y sus columnas en el dataset
METRICAS = {
    'Frecuencia de eventos': 'FRECUENCIA',
    'Familias afectadas':    'FAMILIAS',
    'Personas afectadas':    'PERSONAS',
    'Acueductos afectados':  'ACUED.',
    'Centros de salud':      'C.SALUD',
    'Vías afectadas':        'VIAS',
    'Hectáreas afectadas':   'HECTAREAS',
}
```

4. Carga y Preprocesamiento de Datos

El conjunto de datos de emergencias está en formato **Parquet**, un formato columnar eficiente que permite cargar solo las columnas necesarias y tiene excelente compresión. Contiene los registros oficiales de emergencias de la UNGRD para el período del análogo El Niño 2023–2024.

```
def cargar_base_maestra(ruta: str) -> pd.DataFrame:
    """Carga y filtra los registros del análogo Niño 2023-2024."""
    df = pd.read_parquet(ruta)
```

```
    df = df[df['FECHA_DATETIME'] >= pd.to_datetime('2023-06-01')].copy()
    df['TRIMESTRE_FORMAL'] = df['PERIODO_TRIMESTRAL'].map(TRIM_MAP)
    return df


def cargar_geojsons(ruta_deptos: str, ruta_municipios: str) -> tuple:
    """Carga los GeoJSON y normaliza las claves COD_GEO y NOMBRE_GEO."""
    with open(ruta_deptos, 'r', encoding='utf-8') as f:
        geo_d = json.load(f)
    for feat in geo_d.get('features', []):
        feat['properties']['COD_GEO']    = str(feat['properties'].get('DPTO_CCDGO', '0')).zfill(2)
        feat['properties']['NOMBRE_GEO'] = feat['properties'].get('DPTO_CNMBR', 'Desconocido')

    with open(ruta_municipios, 'r', encoding='utf-8') as f:
        geo_m = json.load(f)
    for feat in geo_m.get('features', []):
        cod = feat['properties'].get('MPIO_CCNCT') or feat['properties'].get('DPMP') or '0'
        feat['properties']['COD_GEO']    = str(cod).zfill(5)
        feat['properties']['NOMBRE_GEO'] = feat['properties'].get('MPIO_CNMBR', 'Desconocido')

    return geo_d, geo_m


df = cargar_base_maestra(RUTA_DATOS)
geo_deptos, geo_municipios = cargar_geojsons(RUTA_DEPTOS, RUTA_MUNICIPIOS)

print('=' * 55)
print('  DATOS CARGADOS — ANÁLOGO EL NIÑO 2023–2024')
print('=' * 55)
print(f'  Registros:        {len(df):,}')
print(f'  Período:          {df["FECHA_DATETIME"].min().date()}  →  {df["FECHA_DATETIME"].max().date()}')
print(f'  Departamentos:    {df["Nombre Departamento"].nunique()}')
print(f'  Municipios:       {df["Nombre Municipio"].nunique()}')
print(f'  Tipos de evento:  {df["EVENTO"].nunique()}')
print('=' * 55)
```

```
=======================================================
  DATOS CARGADOS — ANÁLOGO EL NIÑO 2023–2024
=======================================================
  Registros:        8,169
  Período:          2023-06-01  →  2024-05-31
  Departamentos:    32
  Municipios:       940
  Tipos de evento:  12
=======================================================
```

5. Indicadores de Impacto Globales

Los indicadores de impacto agregados para todo el período del análogo (junio 2023 – mayo 2024) constituyen el **límite superior de referencia** para los escenarios ante un Niño 2026–2027 de magnitud comparable. Representan el peor escenario histórico reciente documentado por la UNGRD.

```
resumen = {
    'Total de eventos':           len(df),
    'Familias afectadas':         int(df['FAMILIAS'].sum()),
    'Personas afectadas':         int(df['PERSONAS'].sum()),
    'Acueductos afectados':       int(df['ACUED.'].sum()),
    'Centros de salud afectados': int(df['C.SALUD'].sum()),
    'Vías afectadas':             int(df['VIAS'].sum()),
    'Hectáreas afectadas':        int(df['HECTAREAS'].sum()),
}

df_kpi = pd.DataFrame(list(resumen.items()), columns=['Indicador', 'Valor'])

(
    df_kpi.style
```

```
    .format({'Valor': '{:,.0f}'})
    .set_caption('Indicadores de impacto — Análogo El Niño 2023–2024 (período completo)')
    .set_table_styles([
        {'selector': 'caption',
         'props': [('font-weight', 'bold'), ('color', UNGRD_BLUE),
                   ('font-size', '13px'), ('text-align', 'left'), ('padding-bottom', '8px')]},
        {'selector': 'th',
         'props': [('background-color', UNGRD_BLUE), ('color', 'white'), ('font-size', '12px')]},
        {'selector': 'td:nth-child(2)',
         'props': [('text-align', 'right'), ('font-weight', 'bold'), ('color', UNGRD_BLUE)]},
    ])
    .hide(axis='index')
)
```

Indicador	Valor
Total de eventos	8,169
Familias afectadas	183,994
Personas afectadas	1,660,604
Acueductos afectados	226
Centros de salud afectados	63
Vías afectadas	768
Hectáreas afectadas	243,345

6. Distribución por Categoría Climática

La clasificación climática de los eventos permite separar la señal directa del Fenómeno El Niño:

- ☀️ **Condiciones secas** (señal directa): incendio forestal, sequía, desabastecimiento de agua, helada, racionamiento.
- 🌧️ **Condiciones lluviosas**: inundación, movimiento en masa, avenida torrencial, vendaval, granizada, tormenta eléctrica.

Aunque El Niño genera predominantemente condiciones secas, la fase de transición del fenómeno puede intensificar las lluvias en ciertas regiones, razón por la que ambas categorías de eventos aparecen en el período analizado.

```
df_clima = (
    df.groupby(['TRIMESTRE_FORMAL', 'CATEGORIA_CLIMA'])
    .size()
    .reset_index(name='Eventos')
    .dropna(subset=['TRIMESTRE_FORMAL'])
)

chart_clima = (
    alt.Chart(df_clima)
    .mark_bar(cornerRadiusEnd=4)
    .encode(
        x=alt.X('TRIMESTRE_FORMAL:N', title='Trimestre',
                sort=ORDEN_TRIMESTRES,
                axis=alt.Axis(labelAngle=-20, labelFontSize=11)),
        y=alt.Y('Eventos:Q', title='Total de eventos', stack=True),
        color=alt.Color(
            'CATEGORIA_CLIMA:N',
            scale=alt.Scale(domain=['Lluviosos', 'Secos'],
                            range=[UNGRD_BLUE, UNGRD_RED]),
            legend=alt.Legend(title='Categoría climática', orient='top')
        ),
        tooltip=[
            alt.Tooltip('TRIMESTRE_FORMAL:N', title='Trimestre'),
            alt.Tooltip('CATEGORIA_CLIMA:N',  title='Categoría'),
            alt.Tooltip('Eventos:Q',           title='Eventos', format=',')
        ]
    )
```

```
    .properties(
        height=320,
        title=alt.TitleParams(
            text='Distribución trimestral de eventos por categoría climática',
            subtitle='Análogo El Niño 2023–2024'
        )
    )
    .configure_axis(labelColor='black', titleColor='black', tickColor='black', domainColor='black')
    .configure_title(color='black')
    .configure_legend(labelColor='black', titleColor='black')
)

chart_clima
```

7. Mapas Coropléticos Departamentales

Los mapas coropléticos muestran la concentración espacial de los eventos y sus impactos a nivel departamental. Se utiliza **clasificación por cuartiles** (más robusta ante distribuciones sesgadas, habituales en datos de emergencias) para determinar los rangos de color.

Los departamentos en el cuartil superior (Q4) son los territorios con **mayor exposición histórica** y donde las acciones de anticipación son más urgentes.

```
def crear_mapa_coroplético(
    df_filtrado: pd.DataFrame,
    geo_data: dict,
    col_agrup: str,
    metrica: str,
    escala_color: str,
    titulo_leyenda: str,
    alias_geo: str = 'Territorio:',
    zoom_start: int = 5,
) -> folium.Map:
    """Genera un mapa coropléticO interactivo con clasificación por cuartiles."""
    if metrica == 'FRECUENCIA':
        agrup = df_filtrado.groupby(col_agrup).size().reset_index(name='Valor')
    else:
        agrup = df_filtrado.groupby(col_agrup)[metrica].sum().reset_index(name='Valor')

    n_digits = 2 if col_agrup == 'Codigo Departamento' else 5
    agrup[col_agrup] = agrup[col_agrup].astype(str).str.zfill(n_digits)

    m = folium.Map(location=[4.5709, -74.2973], zoom_start=zoom_start,
                   tiles='CartoDB positron', control_scale=True)

    m.get_root().html.add_child(folium.Element("""
    <style>
        svg text { fill: #000 !important; font-weight: bold !important; font-size: 10px !important; }
        .legend { transform: scale(0.88); transform-origin: top right;
                  background: rgba(255,255,255,0.9); border-radius: 4px; }
    </style>"""))

    if agrup['Valor'].sum() == 0:
        return m

    bins = sorted(set(agrup['Valor'].quantile([0, 0.25, 0.5, 0.75, 1.0]).tolist()))
    bins = bins if len(bins) >= 4 else 4

    cp = folium.Choropleth(
        geo_data=geo_data, data=agrup,
        columns=[col_agrup, 'Valor'], key_on='feature.properties.COD_GEO',
        fill_color=escala_color, fill_opacity=0.80, line_opacity=0.30,
        legend_name=titulo_leyenda, bins=bins
    ).add_to(m)

    for s in cp.geojson.data['features']:
```

```
        val = agrup.loc[agrup[col_agrup] == s['properties']['COD_GEO'], 'Valor'].values
        s['properties']['Valor'] = int(val[0]) if len(val) > 0 else 0

    folium.GeoJsonTooltip(
        fields=['NOMBRE_GEO', 'Valor'],
        aliases=[alias_geo, f'{titulo_leyenda}:'],
        style='font-size:11px; background:rgba(255,255,255,0.95); border:1px solid #0154a5; border-radius:4px;
color:#000;'
    ).add_to(cp.geojson)

    return m
```

7.1 Frecuencia Total de Eventos

```
mapa_frecuencia = crear_mapa_coroplético(
    df, geo_deptos, 'Codigo Departamento', 'FRECUENCIA',
    'YlOrRd', 'Frecuencia de eventos', 'Departamento:'
)
mapa_frecuencia.save('mapa_frecuencia.html')
display(mapa_frecuencia)
```

Make this Notebook Trusted to load map: File -> Trust Notebook

7.2 Familias Afectadas

```
mapa_familias = crear_mapa_coroplético(
    df, geo_deptos, 'Codigo Departamento', 'FAMILIAS',
    'YlOrBr', 'Familias afectadas', 'Departamento:'
)
mapa_familias.save('mapa_familias.html')
display(mapa_familias)
```

Make this Notebook Trusted to load map: File -> Trust Notebook

7.3 Eventos en Condiciones Secas — Señal Directa del Fenómeno El Niño

```
df_secos = df[df['CATEGORIA_CLIMA'] == 'Secos'].copy()

mapa_secos = crear_mapa_coroplético(
    df_secos, geo_deptos, 'Codigo Departamento', 'FRECUENCIA',
    'OrRd', 'Eventos en condiciones secas', 'Departamento:'
)
mapa_secos.save('mapa_secos.html')
display(mapa_secos)
```

Make this Notebook Trusted to load map: File -> Trust Notebook

8. Ranking Departamental

Los rankings permiten priorizar territorios para las acciones de anticipación. El **Top 10 departamental** por cada métrica identifica los territorios que concentran la mayor parte del impacto histórico y donde la preparación para la respuesta es más crítica.

```
def grafico_top10(
    df_entrada: pd.DataFrame,
    col_nombre: str,
    metrica: str,
    titulo: str,
    esquema: str = 'yelloworangered'
) -> alt.Chart:
    """Barras horizontales con el Top 10 de territorios para una métrica dada."""
    if metrica == 'FRECUENCIA':
        df_top = df_entrada[col_nombre].value_counts().reset_index()
        df_top.columns = ['Territorio', 'Valor']
    else:
        df_top = df_entrada.groupby(col_nombre)[metrica].sum().reset_index()
        df_top.columns = ['Territorio', 'Valor']

    df_top = df_top[df_top['Valor'] > 0].sort_values('Valor', ascending=False).head(10)
```

```
    return (
        alt.Chart(df_top)
        .mark_bar(cornerRadiusEnd=4)
        .encode(
            x=alt.X('Valor:Q', title='Valor'),
            y=alt.Y('Territorio:N', sort='-x', title='',
                    axis=alt.Axis(labelLimit=500, labelFontSize=11)),
            color=alt.Color('Valor:Q', scale=alt.Scale(scheme=esquema), legend=None),
            tooltip=[
                alt.Tooltip('Territorio:N', title='Departamento'),
                alt.Tooltip('Valor:Q',      title='Valor', format=',')
            ]
        )
        .properties(height=300, title=alt.TitleParams(text=titulo))
        .configure_axis(labelColor='black', titleColor='black',
                        tickColor='black',  domainColor='black')
        .configure_title(color='black')
    )

grafico_top10(df, 'Nombre Departamento', 'FRECUENCIA',
              'Top 10 departamentos — Frecuencia total de eventos', 'yelloworangered')

grafico_top10(df, 'Nombre Departamento', 'FAMILIAS',
              'Top 10 departamentos — Familias afectadas', 'yelloworangebrown')

grafico_top10(df_secos, 'Nombre Departamento', 'FRECUENCIA',
              'Top 10 departamentos — Eventos en condiciones secas (señal El Niño)', 'orangered')
```

9. Análisis Temporal

La evolución trimestral permite identificar los **periodos de mayor concentración de emergencias**, información clave para planificar el preposicionamiento de ayudas humanitarias y el despliegue de capacidades de respuesta antes de que el fenómeno alcance su máxima intensidad.

```
df_trim = (
    df.groupby('TRIMESTRE_FORMAL').size().reset_index(name='Eventos')
    .dropna(subset=['TRIMESTRE_FORMAL'])
)

chart_temporal = (
    alt.Chart(df_trim)
    .mark_bar(cornerRadiusEnd=4, color=UNGRD_BLUE)
    .encode(
        x=alt.X('TRIMESTRE_FORMAL:N', title='Trimestre',
                sort=ORDEN_TRIMESTRES, axis=alt.Axis(labelAngle=-20)),
        y=alt.Y('Eventos:Q', title='Total de eventos'),
        tooltip=[
            alt.Tooltip('TRIMESTRE_FORMAL:N', title='Trimestre'),
            alt.Tooltip('Eventos:Q',           title='Eventos', format=',')
        ]
    )
    .properties(
        height=300,
        title=alt.TitleParams(
            text='Distribución trimestral de eventos — Análogo El Niño 2023–2024',
            subtitle='Todos los tipos de evento y categorías climáticas'
        )
    )
    .configure_axis(labelColor='black', titleColor='black', tickColor='black', domainColor='black')
    .configure_title(color='black')
)

chart_temporal

# Evolución del impacto humanitario trimestral con matplotlib
df_trim_impact = (
    df.groupby('TRIMESTRE_FORMAL')
    .agg(Familias=('FAMILIAS', 'sum'), Personas=('PERSONAS', 'sum'))
```

```
    .reset_index()
    .dropna(subset=['TRIMESTRE_FORMAL'])
)
df_trim_impact['Orden'] = df_trim_impact['TRIMESTRE_FORMAL'].map(
    {t: i for i, t in enumerate(ORDEN_TRIMESTRES)}
)
df_trim_impact = df_trim_impact.sort_values('Orden')

fig, axes = plt.subplots(1, 2, figsize=(13, 4))
etiquetas = [t.replace(' ', '\n') for t in df_trim_impact['TRIMESTRE_FORMAL']]

axes[0].bar(etiquetas, df_trim_impact['Familias'], color=UNGRD_BLUE, alpha=0.85, edgecolor='white')
axes[0].set_title('Familias afectadas por trimestre', fontsize=12, fontweight='bold', pad=10)
axes[0].set_ylabel('Familias afectadas')
axes[0].yaxis.set_major_formatter(plt.FuncFormatter(lambda x, _: f'{x:,.0f}'))

axes[1].bar(etiquetas, df_trim_impact['Personas'], color=UNGRD_ACCENT, alpha=0.85, edgecolor='white')
axes[1].set_title('Personas afectadas por trimestre', fontsize=12, fontweight='bold', pad=10)
axes[1].set_ylabel('Personas afectadas')
axes[1].yaxis.set_major_formatter(plt.FuncFormatter(lambda x, _: f'{x:,.0f}'))

plt.tight_layout()
plt.savefig('impacto_trimestral.png', bbox_inches='tight', dpi=150)
plt.show()
print('Figura guardada: impacto_trimestral.png')
```

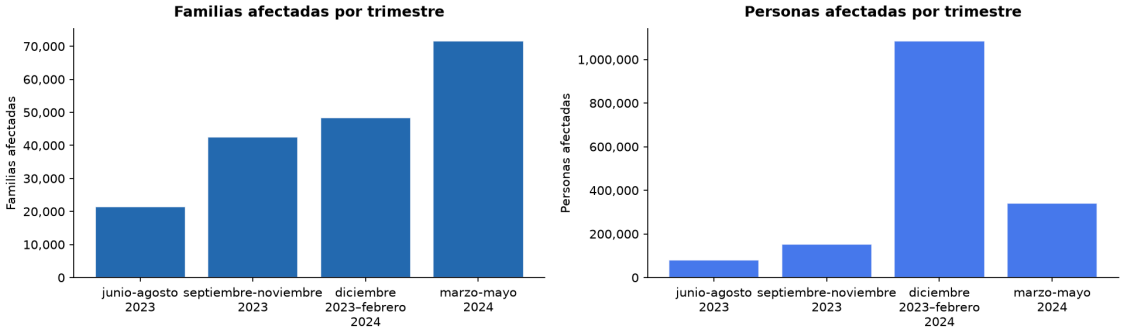

Figure 1: Evolución trimestral del impacto humanitario durante el análogo El Niño 2023–2024. Izquierda: familias afectadas por trimestre. Derecha: personas afectadas por trimestre.

10. Análisis por Tipo de Evento

La desagregación por tipo de amenaza permite orientar la formulación de **planes de contingencia específicos**. Los eventos que aparecen tanto en condiciones lluviosas como secas señalan amenazas de ocurrencia más frecuente durante el año, independientemente de la fase climática.

```
df_tipo = df['EVENTO'].value_counts().reset_index()
df_tipo.columns = ['Tipo de evento', 'Frecuencia']

chart_tipo = (
    alt.Chart(df_tipo.head(15))
    .mark_bar(cornerRadiusEnd=4)
    .encode(
        x=alt.X('Frecuencia:Q', title='Número de eventos'),
        y=alt.Y('Tipo de evento:N', sort='-x', title='',
                axis=alt.Axis(labelLimit=500, labelFontSize=11)),
        color=alt.Color('Frecuencia:Q', scale=alt.Scale(scheme='yelloworangebrown'), legend=None),
        tooltip=[
            alt.Tooltip('Tipo de evento:N', title='Tipo'),
            alt.Tooltip('Frecuencia:Q',     title='Eventos', format=',')
        ]
    )
    .properties(
```

```
        height=420,
        title=alt.TitleParams(
            text='Top 15 tipos de evento — Período completo',
            subtitle='Análogo El Niño 2023–2024'
        )
    )
    .configure_axis(labelColor='black', titleColor='black', tickColor='black', domainColor='black')
    .configure_title(color='black')
)

chart_tipo
```

```
df_tipo_clima = (
    df.groupby(['EVENTO', 'CATEGORIA_CLIMA']).size().reset_index(name='Frecuencia')
)
top_eventos = df_tipo.head(12)['Tipo de evento'].tolist()

chart_tipo_clima = (
    alt.Chart(df_tipo_clima[df_tipo_clima['EVENTO'].isin(top_eventos)])
    .mark_bar(cornerRadiusEnd=4)
    .encode(
        x=alt.X('Frecuencia:Q', title='Número de eventos'),
        y=alt.Y('EVENTO:N', sort='-x', title='',
                axis=alt.Axis(labelLimit=500, labelFontSize=11)),
        color=alt.Color(
            'CATEGORIA_CLIMA:N',
            scale=alt.Scale(domain=['Lluviosos', 'Secos'], range=[UNGRD_BLUE, UNGRD_RED]),
            legend=alt.Legend(title='Categoría climática', orient='top')
        ),
        tooltip=[
            alt.Tooltip('EVENTO:N',          title='Tipo de evento'),
            alt.Tooltip('CATEGORIA_CLIMA:N', title='Categoría'),
            alt.Tooltip('Frecuencia:Q',      title='Eventos', format=',')
        ]
    )
    .properties(
        height=380,
        title=alt.TitleParams(
            text='Tipos de evento por categoría climática (Top 12)',
            subtitle='Análogo El Niño 2023–2024'
        )
    )
    .configure_axis(labelColor='black', titleColor='black', tickColor='black', domainColor='black')
    .configure_title(color='black')
    .configure_legend(labelColor='black', titleColor='black')
)

chart_tipo_clima
```

11. Exportación de Resultados

Las tablas de resultados se exportan en CSV para su uso en sistemas de información territorial, tableros de mando o análisis complementarios.

```
os.makedirs('resultados', exist_ok=True)

ranking_depto = (
    df.groupby(['Codigo Departamento', 'Nombre Departamento'])
    .agg(
        Frecuencia   =('EVENTO',    'count'),
        Familias     =('FAMILIAS',  'sum'),
        Personas     =('PERSONAS',  'sum'),
        Acueductos   =('ACUED.',    'sum'),
        CentrosSalud =('C.SALUD',   'sum'),
        Vias         =('VIAS',      'sum'),
        Hectareas    =('HECTAREAS', 'sum'),
```

```
    )
    .reset_index()
    .sort_values('Frecuencia', ascending=False)
)
ranking_depto.to_csv('resultados/ranking_departamentos.csv', index=False, encoding='utf-8-sig')

ranking_mpio = (
    df.groupby(['Codigo Municipio', 'Nombre Municipio', 'Nombre Departamento'])
    .agg(
        Frecuencia=('EVENTO',   'count'),
        Familias  =('FAMILIAS', 'sum'),
        Personas  =('PERSONAS', 'sum'),
    )
    .reset_index()
    .sort_values('Frecuencia', ascending=False)
)
ranking_mpio.to_csv('resultados/ranking_municipios.csv', index=False, encoding='utf-8-sig')

resumen_trim = (
    df.groupby(['TRIMESTRE_FORMAL', 'CATEGORIA_CLIMA'])
    .agg(Frecuencia=('EVENTO', 'count'), Familias=('FAMILIAS', 'sum'), Personas=('PERSONAS', 'sum'))
    .reset_index()
    .dropna(subset=['TRIMESTRE_FORMAL'])
)
resumen_trim.to_csv('resultados/resumen_trimestral.csv', index=False, encoding='utf-8-sig')

print('✅ Archivos exportados en resultados/:')
for f in sorted(os.listdir('resultados')):
    print(f'   {f}')
```

```
✅ Archivos exportados en resultados/:
   ranking_departamentos.csv
   ranking_municipios.csv
   resumen_trimestral.csv
```

12. Limitaciones y Próximos Pasos

Limitaciones del análisis:

1. **Análogo único**: El análisis se basa en un solo evento histórico (Niño 2023–2024). La variabilidad entre episodios puede ser significativa según la intensidad (moderado/fuerte/muy fuerte) y los patrones de teleconexión.
2. **Subregistro de eventos**: Los datos de la UNGRD capturan emergencias que superaron la capacidad local de respuesta. Eventos de menor magnitud pueden no estar registrados.
3. **Cambios en exposición y vulnerabilidad**: Las condiciones territoriales cambian con el tiempo; los valores del análogo 2023–2024 no incorporan dinámicas posteriores.

Extensiones recomendadas:

Extensión	Descripción	Complejidad
Ensamble de análogos	Incorporar Niño 2009–2010 y 2015–2016 para construir rangos de escenarios	⭐⭐⭐
Índice ONI	Correlacionar intensidad del fenómeno (ONI) con frecuencia de emergencias	⭐⭐
Vulnerabilidad compuesta	Cruzar ranking de impacto con IVA (DANE) para riesgo compuesto	⭐⭐⭐
Escala municipal	Aplicar el mismo flujo a los departamentos priorizados	⭐⭐

Referencias

1. UNGRD (2024). *Reportes de emergencias históricos — Subdirección para el Conocimiento del Riesgo*. Bogotá: UNGRD.
2. IDEAM (2024). *Boletín de alertas hidrometeorológicas*. Bogotá: IDEAM.
3. DANE (2023). *División Político-Administrativa de Colombia — DIVIPOLA*. Bogotá: DANE.

4. NOAA / Climate Prediction Center (2024). *El Niño/Southern Oscillation (ENSO) Diagnostic Discussion*. National Weather Service.
5. UNDRR (2015). *Marco de Sendai para la Reducción del Riesgo de Desastres 2015–2030*. Ginebra: Naciones Unidas.
