

Exploración y Preprocesamiento del Dataset

1. Introducción

Objetivo: construir un conjunto de datos listo para modelado supervisado, orientado a predecir zonas afectadas por inundaciones. A partir de la integración de múltiples fuentes geoespaciales, se realiza un proceso detallado de limpieza, codificación, reprojeción, cálculo de centroides y generación de variables predictoras.

El enfoque combina técnicas de análisis espacial con criterios ambientales y sociales, incorporando variables como la frecuencia de inundación, la distancia a cuerpos y cursos de agua, y la presencia de microbasurales cercanos.

Cada paso se documenta y justifica, con visualizaciones y verificaciones técnicas que aseguran la calidad del dataset final.

Este trabajo sienta las bases para aplicar algoritmos de clasificación en la siguiente etapa del proyecto (Entrega 3), con el fin de aportar valor desde el análisis ambiental y social.

▼ 2. Preparación del entorno Carga de Datasets

```
# Carga manual de archivos
from google.colab import files
uploaded = files.upload()
```

Elegir archivos 4 archivos

```
du_techo_cuerpos_de_agua.geojson(n/a) - 312782 bytes, last modified: 2/10/2025 - 100% done
du_techo_cursos_de_agua.geojson(n/a) - 1850932 bytes, last modified: 2/10/2025 - 100% done
du_techo_microbasurales.geojson(n/a) - 1178997 bytes, last modified: 2/10/2025 - 100% done
du_techo_zonas_inundables.csv(text/csv) - 1491171 bytes, last modified: 2/10/2025 - 100% done
Saving du_techo_cuerpos_de_agua.geojson to du_techo_cuerpos_de_agua.geojson
Saving du_techo_cursos_de_agua.geojson to du_techo_cursos_de_agua.geojson
Saving du_techo_microbasurales.geojson to du_techo_microbasurales.geojson
Saving du_techo_zonas_inundables.csv to du_techo_zonas_inundables.csv
```

```
# Preparamos el entorno
import pandas as pd
import geopandas as gpd

# Cargar los datasets
zonas_inundables = pd.read_csv("du_techo_zonas_inundables.csv")
cuerpos_agua = gpd.read_file("du_techo_cuerpos_de_agua.geojson")
cursos_agua = gpd.read_file("du_techo_cursos_de_agua.geojson")
microbasurales = gpd.read_file("du_techo_microbasurales.geojson")
```

En esta sección se cargan los cuatro datasets que componen la base de datos del proyecto.

du_techo_zonas_inundables.csv contiene información tabular sobre zonas afectadas por inundaciones, incluyendo la variable objetivo se_inunda_.

du_techo_cuerpos_de_agua.geojson, du_techo_cursos_de_agua.geojson y du_techo_microbasurales.geojson son capas geoespaciales en formato GeoJSON que representan elementos ambientales relevantes. Se utilizan pandas para el archivo CSV y geopandas para los archivos GeoJSON.

Todos los archivos fueron obtenidos de fuentes públicas: DU-Techo y RENABAP (01/10/2025), y se subieron manualmente a Colab para su procesamiento. Dataset (fuente)

<https://www.datos.gob.ar/dataset/habitat-factores riesgo-barrios-populares-gran-buenos-aires>

▼ 3. Exploración Inicial y Codificación de la Variable Objetivo

Vamos a analizar como está estructurado el dataset principal (du_techo_zonas_inundables.csv) y preparar la variable se_inunda_ para el modelo.

```
# Exploramos la estructura del dataset
zonas_inundables.info() # vemos estructura general
zonas_inundables.head() # vemos primeras filas
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2746 entries, 0 to 2745
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   WKT          2746 non-null   object  
 1   id           2746 non-null   int64  
 2   id_poligon  2741 non-null   float64 
 3   se_inunda_   2746 non-null   object  
 4   con_que_fr  2746 non-null   object  
 5   provinicia  2746 non-null   object  
 6   departamen  2746 non-null   object  
 7   localidad    2746 non-null   object  
dtypes: float64(1), int64(1), object(6)
memory usage: 171.8+ KB
```

0	MULTIPOLYGON ((-58.2143471291028 -26.21995308...))	1	17325.0	SÍ, SÓLO EN UN SECTOR DEL BARRIO	SOLAMENTE CUANDO DILUVIA (UNA O DOS VECES POR ...	Formosa	For	
1	MULTIPOLYGON ((-58.2179226698168 -26.20925840...))	2	17322.0	SÍ, SÓLO EN UN SECTOR DEL BARRIO	CADA VEZ QUE LLUEVE FUERTE (MUCHAS VECES POR AÑO)	Formosa	For	
2	MULTIPOLYGON ((-58.2334597447517 -26.20322391...))	3	18578.0	SÍ, TODO EL BARRIO	SOLAMENTE CUANDO DILUVIA (UNA O DOS VECES POR ...	Formosa	For	
3	MULTIPOLYGON ((-58.2764032110572 -26.27741542...))	4	12001.0	SÍ, SÓLO EN UN SECTOR DEL BARRIO	SOLAMENTE CUANDO DILUVIA (UNA O DOS VECES POR ...	Formosa	For	
4	MULTIPOLYGON ((-58.2777912542224 -26.27851061...))	5	12000.0	SÍ, SÓLO EN UN SECTOR DEL BARRIO	SOLAMENTE CUANDO DILUVIA (UNA O DOS VECES POR ...	Formosa	For	

Próximos pasos: [Generar código con zonas_inundables](#) [New interactive sheet](#)

```
# Vemos distribución de la variable objetivo
# (incluyendo si hay celdas con valores nulos)
```

```
zonas_inundables['se_inunda_'].value_counts(dropna=False)
```

se_inunda_	count
SÍ, TODO EL BARRIO	1374
SÍ, SÓLO EN UN SECTOR DEL BARRIO	1372

dtype: int64

Se utiliza `value_counts(dropna=False)` para contar todas las categorías presentes en la variable `se_inunda_`, incluyendo los valores faltantes (NaN). Esto permite detectar si hay registros incompletos que podrían afectar el modelado o requerir imputación o descarte.

```
# Inspeccionamos el nombre de todas las columnas del dataset
zonas_inundables.columns
```

```
Index(['WKT', 'id', 'id_poligon', 'se_inunda_', 'con_que_fr', 'provincia',
       'departamen', 'localidad'],
      dtype='object')
```

Se inspeccionan las columnas del dataset `zonas_inundables` para identificar las variables disponibles.

Las columnas incluyen:

identificadores (id, id poligon),

información geográfica (provincia, departamen, localidad), y

variables relacionadas con el riesgo de inundación (`se_inunda_`, `con_que_fr`).

la columna WKT, contiene geometrías en formato texto

```
# Exploramos los valores únicos de las columnas categóricas
# Imprime los valores únicos
for col in zonas_inundables.columns:
    if zonas_inundables[col].dtype == 'object':
        print(f"\nValores únicos en {col}:")
        print(zonas_inundables[col].unique())
```

'El Alcázar' 'Puerto Rico' 'San Salvador De Jujuy' 'Monterrico' 'Palpalá'
 'San Pedro De Jujuy' 'Humahuaca' 'Libertador General San Martín'
 'Los Lapachos' 'Aguas Calientes' 'Maimará' 'La Quiaca' 'Caimancito'
 'Rawson' 'Trelew' 'Puerto Madryn' 'Comodoro Rivadavia' 'Concepción'
 'Lago Puelo' 'Neuquén' 'Centenario' 'Plottier' 'Junín De Los Andes'
 'Aluminé' 'Senillosa' 'San Miguel De Tucumán' 'Ingenio La Florida'
 'Alderetes' 'Banda Del Río Salí' 'Aguilares' 'Famaillá'
 'Juan Bautista Alberdi' 'San Isidro De Lules' 'Acheral' 'El Manantial'
 'Yerba Buena - Marcos Paz' 'Ingenio San Pablo' 'Villa Carmela'
 'Las Talitas' 'Alto Verde' 'Arcadia' 'San Roque' 'La Reducción'
 'Diagonal Norte, Luz Y Fuerza, Los Pocitos, Villa Nueva Italia'
 'Tafí Viejo' 'Barrio San Felipe' 'Río Chico' 'Garmendia'
 'Villa Burruyacú' 'Ex Ingenio Los Ralos' 'Río Colorado' 'Santa Lucía'
 'Santa Ana' 'Simoca' 'Amaicha Del Valle' 'Tafí Del Valle'
 'Villa De Trancas' 'Córdoba Capital' 'Estacion Juárez Celman'
 'Villa Allende' 'Alta Gracia' 'Oncativo' 'Unquillo' 'Jesús María'
 'Morteros' 'Cosquín' 'Villa Carlos Paz' 'Río Cuarto' 'Bouwer'
 'Río Tercero' 'Bell Ville' 'Villa Nueva' 'La Calera' 'Montecristo'
 'Cruz Del Eje' 'Villa Giardino' 'Río Segundo' 'Santa Ana De Los Guácaras'
 'Corrientes' 'Laguna Brava' 'Saladas' 'Goya' 'Esquina'
 'San Luis Del Palmar' 'Paso De Los Libres' 'Concordia' 'Santo Tomé'
 'Empedrado' 'Mburucuyá' 'Monte Caseros' 'Paraná' 'Colonia Avellaneda'
 'Concepción Del Uruguay' 'Gualeguay' 'Gualeguaychú' 'Nogoyá' 'Victoria'
 'Rosario Del Tala' 'Federación' 'Colón' 'La Paz' 'Rincón de Nogoyá'
 'Diamante' 'Villaguay' 'Los Corralitos' 'Las Heras' 'Panquehua'
 'El Borbollón' 'Mendoza' 'Guaymallén' 'Fray Luis Beltrán' 'Russell'
 'Palmira' 'Villa Tulumaya' 'Rodeo Del Medio' 'Tunuyán' 'General Alvear'
 'El Plumerillo' 'Sarmiento' 'Mayor Drummond' 'Trapiche' 'Buena Nueva'
 'Bermejo' 'El Challao' 'Chacras De Coria' 'Jesús Nazareno'
 'Rodeo De La Cruz' 'Perdriel' 'Colonia Segovia'
 'Vertientes del Pedemonte' 'San Martín' 'Rivadavia' 'La Arboleda'
 'Malargüe' 'San Rafael' 'El Peral' 'Cuadro Benegas' 'Las Paredes' 'Maipú'
 'Eugenio Bustos' 'Colonia Molina' 'El Sauce' 'Ugarteche' 'Agrelo'
 'Uspallata' 'Vista Flores' 'Puente De Hierro' 'Goudge' 'Cipolletti'
 'Allen' 'General Roca' 'Cinco Saltos' 'Balsa Las Perlas'
 'San Carlos De Bariloche' 'El Bolsón' 'Viedma' 'Villa Regina'
 'San Antonio Oeste' 'Mainqué' 'Las Grutas' 'Cervantes' 'Villa Mercedes'
 'San Luis' 'Juana Koslay' 'Rosario' 'Villa Gobernador Gálvez' 'Pérez'
 'Granadero Baigorria' 'Alvear' 'Capitán Bermúdez' 'Roldán' 'San Lorenzo'
 'Pueblo Esther' 'General Lagos' 'Arroyo Seco' 'Sauce Viejo'
 'Santa Rosa De Calchines' 'San José Del Rincón' 'Santa Fe' 'Arroyo Leyes'
 'Venado Tuerto' 'Calchaquí' 'Casilda' 'Coronda' 'Esperanza' 'Frontera'
 'Laguna Paiva' 'Las Toscas' 'Rafaela' 'San Cristóbal' 'San Jorge'
 'Tostado' 'Villa Constitución' 'Villa Ocampo' 'Reconquista' 'Correa'
 'Médano De Oro' 'Totoras' 'Tacuarendí' 'Piñero' 'Barrancas'
 'Cañada De Gómez' 'Oliveros' 'Cayastá' 'Helvecia' 'Ingeniero Chanourdie'
 'Chimbas' 'La Bebida' 'Caucete' 'Pie De Palo' 'Las Talas' 'San Juan'
 'Ullum' 'Villa Borjas' 'Villa Santa Rosa' 'Vallecito' 'Tres Esquinas'
 'Villa Media Agua' 'Campo De Herrera' 'Firmat'

Interpretación de la exploración inicial

▼ Variable se_inunda_

Valores únicos encontrados:

"SÍ, TODO EL BARRIO" (1374 registros)

"SÍ, SÓLO EN UN SECTOR DEL BARRIO" (1372 registros)

No se detectaron registros con la categoría "NO SE INUNDA", lo cual indica que el dataset está compuesto exclusivamente por zonas que presentan algún nivel de inundación. Es por ello que tomamos la decisión de trabajar con una clasificación binaria supervisada, diferenciando entre inundación total (2) e inundación parcial (1).

Variable con_que_fr

Valores únicos encontrados:

"CADA VEZ QUE LLUEVE FUERTE (MUCHAS VECES POR AÑO)" (1488)

"OCASIONALMENTE (ALGUNAS VECES POR AÑO)" (638)

"SOLAMENTE CUANDO DILUVIA (UNA O DOS VECES POR AÑO)" (572)

"Otro, especificar" (43)

"NS/NC" (2)

Otros valores con muy baja frecuencia (1–2 registros)

Esta variable aporta información sobre la frecuencia de inundación, lo cual puede ser útil como variable predictora. Se observa una predominancia de zonas que se inundan con frecuencia alta. Podríamos agrupar en categorías (Alta, Media, Baja) para facilitar el modelado.

Variable provincia, departamento y localidad

Valores únicos encontrados: Más de 20 provincias. Más de 50 departamentos. Más de 100 localidades. En los 3 casos, aunque las variables aportan contexto geográfico, se decide no incluirlas como variable predictora debido a que:

Tienen muchas categorías.

La ubicación ya está representada por coordenadas (latitud, longitud).

Podría introducir ruido en el modelo y sobreajuste.

Variable WKT

Valores únicos: Geometrías en formato MULTIPOLYGON

Esta columna contiene la geometría de cada zona en formato texto. No se usará directamente en el modelo, pero se puede utilizar para calcular variables derivadas como el centroide o el área.

```
# Vemos la distribución de la variable con_que_fr  
zonas_inundables['con_que_fr'].value_counts(dropna=False)
```

	count
con_que_fr	
CADA VEZ QUE LLUEVE FUERTE (MUCHAS VECES POR AÑO)	1488
OCASIONALMENTE (ALGUNAS VECES POR AÑO)	638
SOLAMENTE CUANDO DILUVIA (UNA O DOS VECES POR AÑO)	572
Otro, especificar	43
NS/NC	2
CADA VEZ QUE LLUEVE (MUCHAS VECES POR AÑO)	2
CADA VEZ QUE LLUEVE FUERTE (MUCHAS VECES AL AÑO)	1

dtype: int64

▼ Interpretación

La mayoría de las zonas se inundan frecuentemente: más de la mitad (1488 de 2746) reportan que se inundan cada vez que llueve fuerte.

Hay una proporción significativa que se inunda ocasionalmente (638) o solo en eventos extremos (572).

Los valores "Otro, especificar" y "NS/NC" son pocos, pero podrían representar respuestas abiertas o falta de información.

Aparecen duplicados con pequeñas variaciones en la redacción, como "CADA VEZ QUE LLUEVE FUERTE (MUCHAS VECES POR AÑO)" y "CADA VEZ QUE LLUEVE FUERTE (MUCHAS VECES AL AÑO)", vamos a normalizar estos textos.

```
# Preprocesamiento (agrupar en tres categorías simplificadas)
# Transformación de variable categórica textual (con_que_fr) en una versión más
def codificar_frecuencia(valor):
    valor = str(valor).upper()
    if "CADA VEZ" in valor or "MUCHAS VECES" in valor:
        return "Alta"
    elif "OCASIONALMENTE" in valor:
        return "Media"
    elif "SOLAMENTE" in valor or "UNA O DOS VECES" in valor:
        return "Baja"
    else:
        return "Otro"

zonas_inundables['frecuencia_codificada'] = zonas_inundables['con_que_fr'].apply
    zonas_inundables['frecuencia_codificada'].value_counts(dropna=False)
```

count	
frecuencia_codificada	
Alta	1491
Media	638
Baja	572
Otro	45

dtype: int64

Se agrupan los valores en cuatro categorías:

Alta: zonas que se inundan cada vez que llueve fuerte o muchas veces por año.

Media: zonas que se inundan ocasionalmente.

Baja: zonas que solo se inundan en eventos extremos.

Otro: respuestas abiertas o no clasificables.

Esta transformación permite reducir la variabilidad textual y facilita el uso de la variable como predictora en el modelo supervisado.

Resultados de la exploración:

Tras analizar las columnas `se_inunda_y` y `con_que_fr`, se observa que el dataset no contiene registros con la categoría "NO SE INUNDA". Esto indica que el archivo `du_techo_zonas_inundables.csv` está compuesto exclusivamente por zonas afectadas por inundaciones, ya sea total o parcialmente. Por lo tanto, se ajusta la variable objetivo a una clasificación binaria:

1: Inundación parcial

2: Inundación total

Esta decisión permite construir un modelo supervisado que prediga el grado de afectación en zonas vulnerables, cumpliendo con el objetivo del proyecto.

Además, se analiza la variable `con_que_fr`, que describe la frecuencia con la que se inunda cada zona. Para facilitar su uso como variable predictora, se realiza un preprocesamiento que agrupa sus valores textuales en cuatro categorías simplificadas:

Alta: Zonas que se inundan cada vez que llueve fuerte o muchas veces por año

Media: Zonas que se inundan ocasionalmente

Baja: Zonas que solo se inundan en eventos extremos

Otro: Respuestas abiertas o no clasificables

Esta nueva variable, `frecuencia_codificada`, se conservará como predictor en el modelo, ya que aporta información complementaria al tipo de inundación.

```
# Codificamos la variable objetivo en formato binario
zonas_gdf['se_inunda_binaria'] = zonas_gdf['se_inunda_'].apply(
    lambda x: 2 if 'TODO EL BARRIO' in str(x).upper() else 1
)

# Verificamos la distribución
zonas_gdf['se_inunda_binaria'].value_counts()
```

	count
se_inunda_binaria	
2	1374
1	1372

dtype: int64

Codificación de la variable objetivo

Se transforma la variable `se_inunda_` en una versión binaria (`se_inunda_binaria`) para facilitar el modelado supervisado.

Se asigna el valor **2** a zonas con inundación total ("TODO EL BARRIO") y **1** a zonas con inundación parcial ("SÓLO EN UN SECTOR").

Esta codificación permite entrenar modelos de clasificación binaria que predigan el grado de afectación por inundaciones.

▼ 4. Construcción de Variables Predictoras Geoespaciales

Incorporamos variables espaciales que puedan influir en el riesgo de inundación, como la proximidad a cuerpos de agua y a cursos de agua (distancias) o la presencia de microbasurales (cantidad). Estas variables se extraen a partir de las capas geoespaciales disponibles en formato GeoJSON.

Justificación del uso de geopandas y shapely Para enriquecer el dataset con variables ambientales, se utilizan capas geoespaciales que contienen información sobre cuerpos de agua, cursos de agua y microbasurales. Estas capas están en formato GeoJSON, lo que permite trabajar con geometrías mediante la librería geopandas, que extiende pandas para datos espaciales. Además, se utiliza shapely, una librería especializada en operaciones geométricas, como el cálculo de distancias entre polígonos y puntos. El uso

de estas herramientas está justificado por la necesidad de generar variables predictoras basadas en la ubicación y el entorno físico de cada zona.

```
# Convertimos zonas_inundables en un GeoDataFrame
import geopandas as gpd
from shapely import wkt

# Convertimos la columna WKT en geometrías
zonas_inundables['geometry'] = zonas_inundables['WKT'].apply(wkt.loads)

# Creamos el GeoDataFrame
zonas_gdf = gpd.GeoDataFrame(zonas_inundables, geometry='geometry')
```

wkt.loads convierte el texto WKT en un objeto geométrico de shapely.

geometry='geometry' le indica a geopandas cuál es la columna que contiene las geometrías.

```
# Verificamos que se creó correctamente
zonas_gdf.head()
```

	WKT	id	id_poligon	se_inunda_	con_que_fr	provinicia	depar...
0	MULTIPOLYGON (((-58.2143471291028 -26.21995308...))	1	17325.0	SÍ, SÓLO EN UN SECTOR DEL BARRIO	SOLAMENTE CUANDO DILUVIA (UNA O DOS VECES POR ...	Formosa	For...
1	MULTIPOLYGON (((-58.2179226698168 -26.20925840...))	2	17322.0	SÍ, SÓLO EN UN SECTOR DEL BARRIO	CADA VEZ QUE LLUEVE FUERTE (MUCHAS VECES POR AÑO)	Formosa	For...
2	MULTIPOLYGON (((-58.2334597447517 -26.20322391...))	3	18578.0	Sí, TODO EL BARRIO	SOLAMENTE CUANDO DILUVIA (UNA O DOS VECES POR ...	Formosa	For...
3	MULTIPOLYGON (((-58.2764032110572 -26.27741542...))	4	12001.0	SÍ, SÓLO EN UN SECTOR DEL BARRIO	SOLAMENTE CUANDO DILUVIA (UNA O DOS VECES POR ...	Formosa	For...
4	MULTIPOLYGON (((-58.2777912542224 -26.27851061...))	5	12000.0	Sí, SÓLO EN UN SECTOR DEL BARRIO	SOLAMENTE CUANDO DILUVIA (UNA O DOS VECES POR ...	Formosa	For...

Próximos pasos: [Generar código con zonas_gdf](#)

[New interactive sheet](#)

La columna geometry fue generada a partir de la columna WKT, utilizando la función wkt.loads para convertir el texto en objetos geométricos tipo MultiPolygon.

Esta transformación permite aplicar operaciones espaciales como cálculo de centroides, distancias y relaciones topológicas, necesarias para construir variables predictoras basadas en el entorno físico de cada zona.

Calculamos centroides de cada polígono

Necesitamos obtener un punto representativo de cada zona (su centroide) para poder calcular distancias espaciales de forma precisa.

Calculamos el centroide de cada polígono utilizando el método `.centroid` de geopandas. El centroide representa el punto medio de cada zona afectada por inundaciones, y se utilizará como referencia para calcular distancias a elementos ambientales cercanos, como cuerpos de agua y microbasurales. Esta operación es posible gracias a la conversión previa del dataset en un GeoDataFrame, que permite aplicar funciones geométricas sobre los objetos espaciales.

▼ Reproyección y cálculo de centroides

Para calcular correctamente los centroides de cada polígono en unidades métricas, se reprojeta el GeoDataFrame al sistema de coordenadas proyectado **EPSG:5347 (POSGAR 2007 / Argentina 3)**, que cubre todo el territorio nacional.

Luego se calcula el centroide de cada geometría, lo cual permite generar variables espaciales como distancia a cuerpos de agua o microbasurales.

```
# Definimos CRS original si no estaba definido
zonas_gdf = zonas_gdf.set_crs(epsg=4326)

# Reproyectamos a sistema en metros (EPSG:5347)
zonas_gdf = zonas_gdf.to_crs(epsg=5347)

# Calculamos centroides
zonas_gdf['centroide'] = zonas_gdf.geometry.centroid

# Verificamos los resultados
zonas_gdf[['geometry', 'centroide']].head()
```

	geometry	centroide	
0	MULTIPOLYGON (((5678459.101 7099534.297, 56785...	POINT (5678606.905 7099622.752)	
1	MULTIPOLYGON (((5678117.978 7100724.445, 56779...	POINT (5678028.389 7101126.919)	
2	MULTIPOLYGON (((5676573.859 7101414.459, 56766...	POINT (5676471.255 7101319.784)	
3	MULTIPOLYGON (((5672171.267 7093250.13, 567225...	POINT (5672170.481 7093118.504)	
4	MULTIPOLYGON (((5672030.974 7093130.605, 56720...	POINT (5672029.439 7093033.059)	

Este bloque nos asegura que:

El CRS original está bien definido (EPSG:4326)

La reproyección se aplica correctamente

Los centroides se calculan en metros y son válidos

```
# Visualización de centroides sobre polígonos
import matplotlib.pyplot as plt

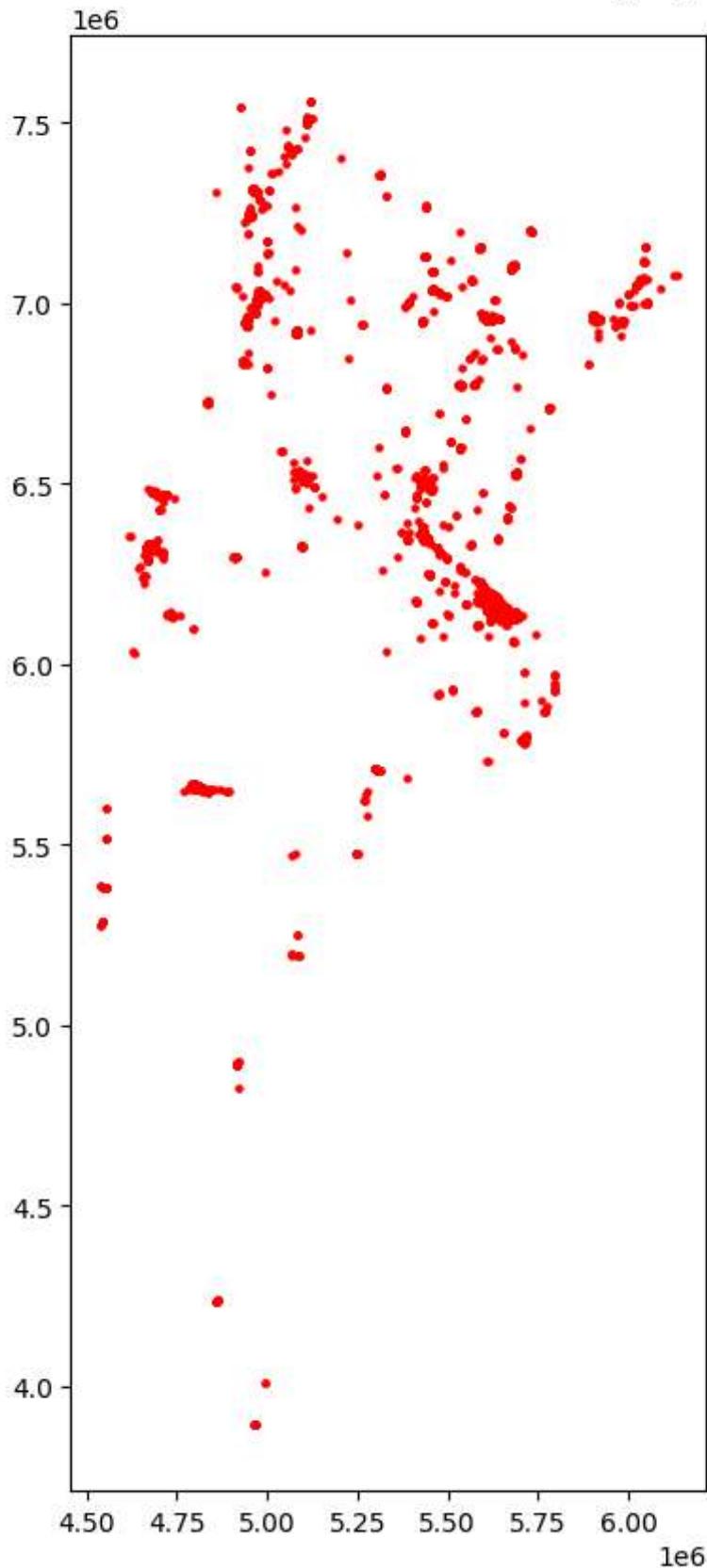
# Creamos el gráfico
fig, ax = plt.subplots(figsize=(10, 10))

# Dibujamos los polígonos (zonas)
zonas_gdf.plot(ax=ax, edgecolor='gray', facecolor='none')

# Dibujamos los centroides en rojo
zonas_gdf.set_geometry('centroide').plot(ax=ax, color='red', markersize=5)

# Título del gráfico
plt.title("Visualización de centroides sobre zonas geográficas")
plt.show()
```

Visualización de centroides sobre zonas geográficas



- Con este gráfico validamos:

Que los centroides estén dentro de cada polígono

Que la reproyección a metros fue exitosa

Que no hay geometrías vacías ni errores de cálculo

Esta transformación es necesaria para obtener centroides y distancias precisas. Una vez reproyectado, se recalcularán los centroides de cada polígono, que serán utilizados como referencia para construir variables predictoras basadas en distancia.

Se visualizan los primeros valores de la columna centroide, que contiene los puntos centrales de cada polígono de zona inundable.

Cada entrada es un objeto POINT con coordenadas en metros, que se utilizará como referencia para calcular distancias a elementos ambientales cercanos, como cuerpos de agua y microbasurales.

Calcular distancia al cuerpo de agua más cercano

Crear una nueva variable predictora que indique qué tan cerca está cada zona de un cuerpo de agua, lo cual puede influir en el riesgo de inundación. Verificamos que la capa cuerpos_agua esté en el mismo sistema de coordenadas que zonas_gdf. sino, reproyectamos.

```
# Reproyectamos cuerpos_agua al mismo CRS que zonas_gdf
cuerpos_agua = cuerpos_agua.to_crs(zonas_gdf.crs)
```

```
# Reproyectar cuerpos de agua al mismo CRS
cuerpos_agua = cuerpos_agua.to_crs(zonas_gdf.crs)

# Calcular distancia mínima desde cada centroide
zonas_gdf['distancia_cuerpo_agua'] = zonas_gdf['centroide'].apply(
    lambda punto: cuerpos_agua.distance(punto).min()
)

# Verificar resultados
zonas_gdf[['centroide', 'distancia_cuerpo_agua']].head()
```

	centroide	distancia_cuerpo_agua	
0	POINT (5678606.905 7099622.752)	1197.382158	
1	POINT (5678028.389 7101126.919)	261.021717	
2	POINT (5676471.255 7101319.784)	1762.778477	
3	POINT (5672170.481 7093118.504)	945.972415	
4	POINT (5672029.439 7093033.059)	1108.288575	

▼ Interpretación del cálculo de distancia a cuerpos de agua

Calculamos la distancia mínima entre el centroide de cada zona inundable y el cuerpo de agua más cercano.

Primero se reproyecta la capa `cuerpos_agua` al mismo sistema de coordenadas que `zonas_gdf` (EPSG:5347), lo que garantiza que las distancias se midan correctamente en metros.

Luego, se aplica una función que recorre cada centroide y calcula su distancia más corta a cualquier geometría de la capa de cuerpos de agua.

El resultado se guarda en una nueva columna llamada `distancia_cuerpo_agua`, que se incorpora como variable predictora en el modelo supervisado.

Esta variable permite evaluar si la proximidad a cuerpos de agua influye en el riesgo de inundación de cada zona.

Calcular distancia al curso de agua más cercano

Crear una nueva variable predictora que indique la distancia entre cada zona inundable y el curso de agua más cercano (ríos, arroyos, canales), que también puede influir en el riesgo de inundación.

```
# Reproyectar cursos de agua al mismo CRS
cursos_agua = cursos_agua.to_crs(zonas_gdf.crs)

# Calcular distancia mínima desde cada centroide
zonas_gdf['distancia_curso_agua'] = zonas_gdf['centroide'].apply(
    lambda punto: cursos_agua.distance(punto).min()
)
```

Calculamos la distancia mínima entre el centroide de cada zona inundable y el curso de agua más cercano (ríos, arroyos, canales).

Primero se reproyecta la capa `cursos_agua` al mismo sistema de coordenadas que `zonas_gdf` (EPSG:5347), lo que garantiza que las distancias se midan correctamente en metros.

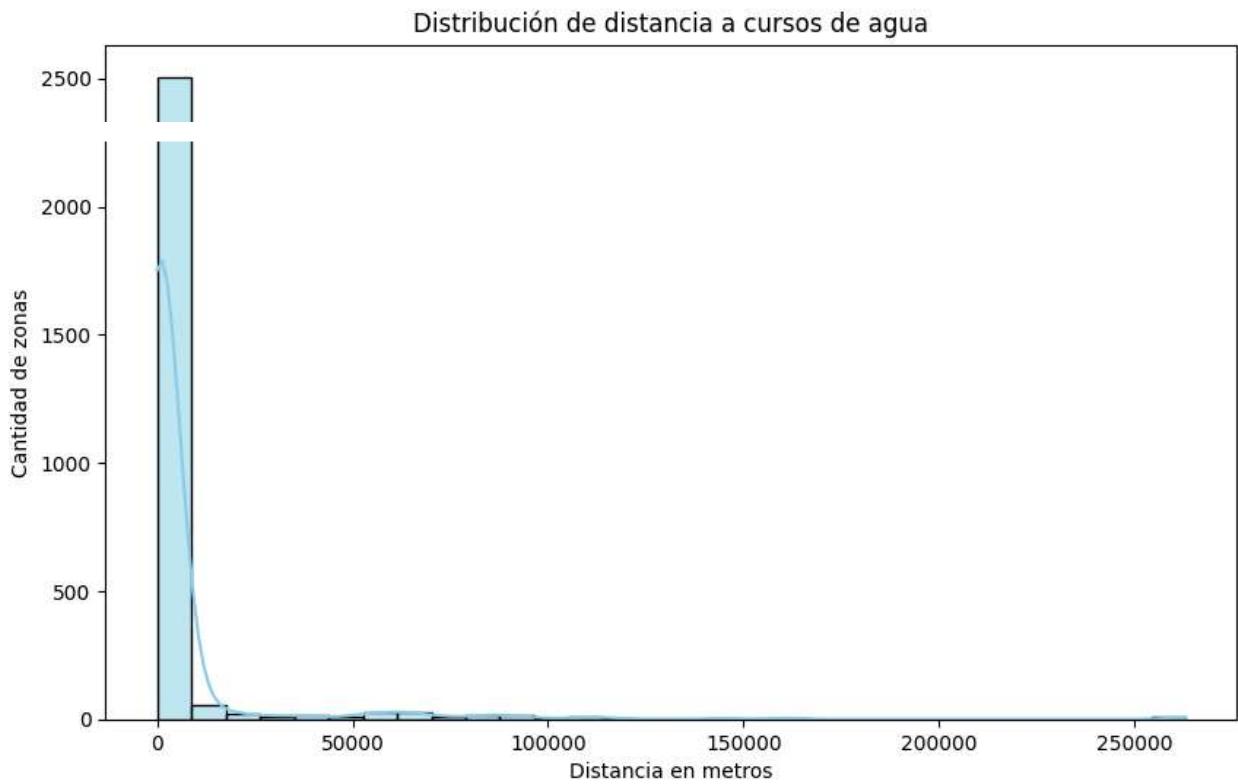
Luego, se aplica una función que recorre cada centroide y calcula su distancia más corta a cualquier geometría de la capa de cursos de agua.

El resultado se guarda en la columna `distancia_curso_agua`, que se incorpora como variable predictora en el modelo supervisado.

Esta variable permite evaluar si la cercanía a cursos de agua está asociada a un mayor riesgo de inundación.

```
# Visualizar distribución
import seaborn as sns
plt.figure(figsize=(10, 6))
sns.histplot(zonas_gdf['distancia_curso_agua'], bins=30, kde=True, color='skyblue')
```

```
plt.title("Distribución de distancia a cursos de agua")
plt.xlabel("Distancia en metros")
plt.ylabel("Cantidad de zonas")
plt.show()
```



Interpretación del gráfico

El gráfico muestra la distribución de la distancia entre cada zona inundable y el curso de agua más cercano. Se observa una concentración de zonas en rangos de distancia relativamente bajos, lo que indica que muchas áreas afectadas por inundaciones se encuentran cerca de cursos de agua. Esta proximidad puede estar asociada a un mayor riesgo de afectación, por lo que la variable `distancia_curso_agua` se considera relevante como predictora en el modelo supervisado. La presencia de una cola más larga hacia la derecha sugiere que también existen zonas más alejadas, lo que podría aportar variabilidad útil para el aprendizaje del modelo.

Conteo de microbasurales cercanos

Crear una nueva variable que indique cuántos microbasurales hay en un radio determinado (por ejemplo, 500 metros) alrededor de cada zona inundable.

```
# Reproyectar microbasurales al mismo CRS
microbasurales = microbasurales.to_crs(zonas_gdf.crs)

# Crear buffer de 500 metros alrededor de cada zona
zonas_gdf['buffer_500m'] = zonas_gdf.geometry.buffer(500)

# Contar microbasurales dentro del buffer
```

```

zonas_gdf['microbasurales_cercanos'] = zonas_gdf['buffer_500m'].apply(
    lambda zona: microbasurales.intersects(zona).sum()
)

# Verificar resultados
zonas_gdf[['microbasurales_cercanos']].describe()

```

microbasurales_cercanos	
count	2746.000000
mean	2.453751
std	3.861662
min	0.000000
25%	0.000000
50%	1.000000
75%	3.000000
max	41.000000

Calculamos cuantos microbasurales se encuentran dentro de un radio de 500 metros alrededor de cada zona inundable.

Para ello, se crea un buffer circular de 500 metros alrededor de cada polígono de `zonas_gdf`, y se verifica cuántos puntos de la capa `microbasurales` intersectan ese buffer.

El resultado se guarda en la columna `microbasurales_cercanos`, que representa la cantidad de focos ambientales cercanos a cada zona.

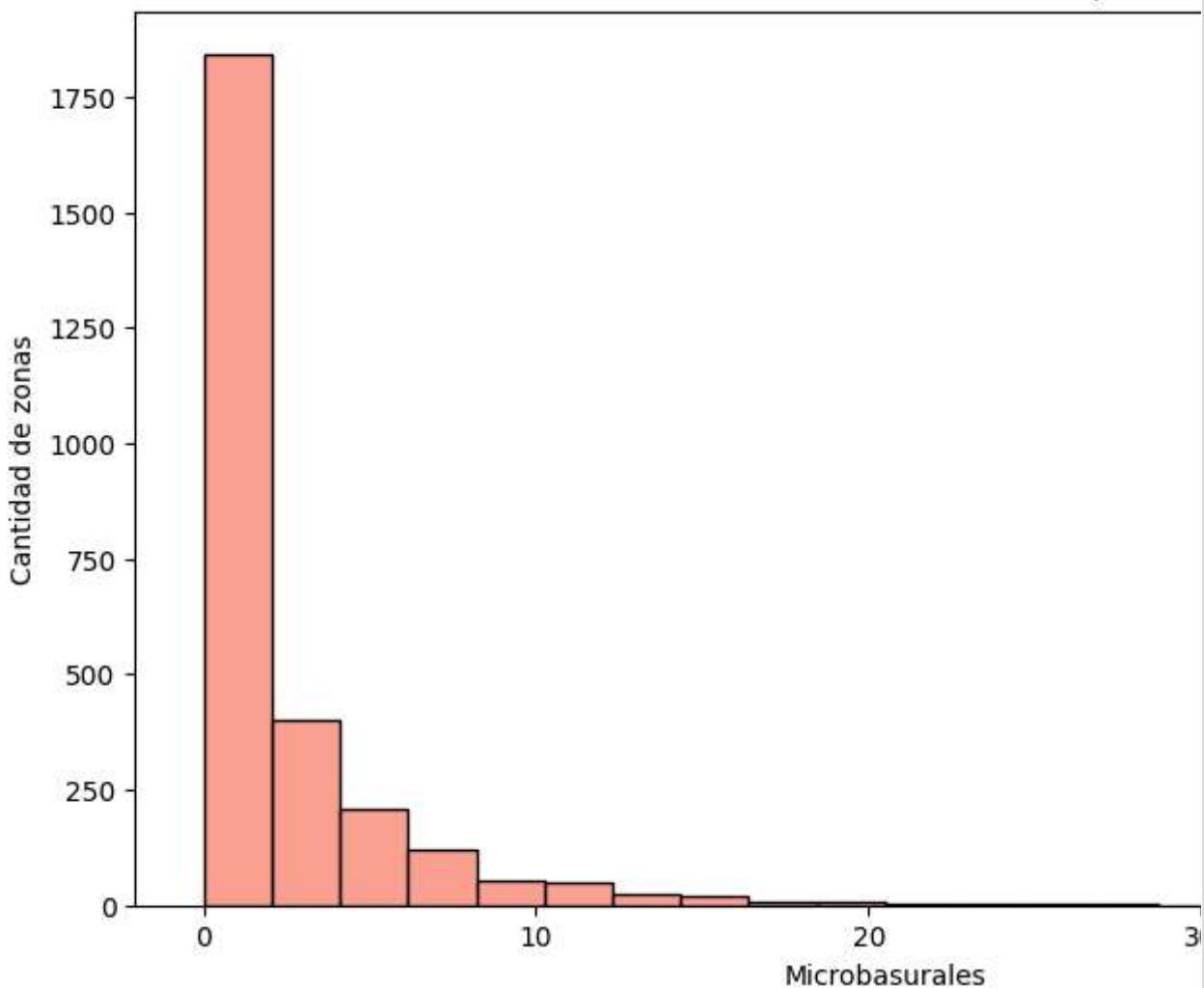
Esta variable se incorpora como predictor en el modelo supervisado, ya que la presencia de microbasurales puede estar relacionada con condiciones de vulnerabilidad ambiental.

```

#visualizamos como se distribuye esta variable
plt.figure(figsize=(10, 6))
sns.histplot(zonas_gdf['microbasurales_cercanos'], bins=20, kde=False, color='salmon')
plt.title("Cantidad de microbasurales cercanos (radio 500m)")
plt.xlabel("Microbasurales")
plt.ylabel("Cantidad de zonas")
plt.show()

```

Cantidad de microbasurales cercanos (radio 500)



▼ Tabla resumen de variables predictoras

Se construye un conjunto de variables predictoras que combinan información categórica y espacial, con el objetivo de modelar el grado de afectación por inundaciones en zonas vulnerables.

Las variables incluyen:

- **se_inunda_binaria**: variable objetivo codificada en formato binario (1: inundación parcial, 2: inundación total)
- **frecuencia_codificada**: frecuencia de inundación agrupada en categorías (Alta, Media, Baja, Otro)
- **distancia_cuerpo_agua**: distancia en metros desde el centroide de cada zona al cuerpo de agua más cercano
- **distancia_curso_agua**: distancia en metros al curso de agua más cercano (ríos, arroyos, canales)
- **microbasurales_cercanos**: cantidad de microbasurales dentro de un radio de 500 metros

Cada variable fue obtenida mediante técnicas de preprocesamiento y análisis geoespacial, y se justifica su inclusión por su relevancia ambiental y social en el fenómeno de estudio.

Este conjunto de variables permite enriquecer el modelo supervisado con información contextual clave, mejorando su capacidad predictiva y su interpretación desde una perspectiva territorial.



Tabla resumen de variables predictoras

Se construye un conjunto de variables que combinan información categórica y espacial, con el objetivo de modelar el grado de afectación por inundaciones en zonas vulnerables. Las variables incluyen:

Variable	Tipo	Fuente / Método de obtención	
se_inunda_binaria	Objetivo	Codificación binaria de la variable <code>se_inunda_</code> (1: parcial, 2: total)	Variab
frecuencia_codificada	Categórica	Agrupación de respuestas textuales de <code>con_que_fr</code>	Indica
distancia_cuerpo_agua	Numérica	Distancia en metros desde el centroide al cuerpo de agua más cercano	Proxin
distancia_curso_agua	Numérica	Distancia en metros desde el centroide al curso de agua más cercano	Influye
microbasurales_cercanos	Numérica	Conteo de microbasurales en un radio de 500 metros	Impac

▼ 6 Preparar el dataset para modelado supervisado

Objetivo Unir todas las variables en un único DataFrame, asegurarse de que no haya valores nulos, y dejarlo listo para entrenar un modelo con scikit-learn.

```
# Seleccionamos las columnas relevantes
variables_modelo = [
    'se_inunda_binaria',
    'frecuencia_codificada',
    'distancia_cuerpo_agua',
    'distancia_curso_agua',
    'microbasurales_cercanos'
]
```