

# Funcionamiento Interno y Aplicaciones de GeoPandas

Dante Márquez López

27 de febrero de 2026

## Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Estructura Fundamental: GeoDataFrame</b>	<b>2</b>
2.1. Tipos de geometrías . . . . .	2
<b>3. Arquitectura Interna</b>	<b>2</b>
3.1. Shapely . . . . .	2
3.2. PyProj . . . . .	3
3.3. Fiona . . . . .	3
<b>4. Sistema de Referencia Espacial (CRS)</b>	<b>3</b>
<b>5. Operaciones Espaciales</b>	<b>3</b>
5.1. Buffer . . . . .	3
5.2. Spatial Join . . . . .	3
5.3. Overlay . . . . .	3
<b>6. Modelo Conceptual</b>	<b>4</b>
<b>7. Aplicaciones en Ciencia de Datos</b>	<b>4</b>
<b>8. Conclusión</b>	<b>4</b>

# 1. Introducción

GeoPandas es una librería de Python diseñada para el análisis y manipulación de datos geoespaciales. Extiende la estructura de `pandas` incorporando una columna geométrica que permite trabajar con objetos espaciales como puntos, líneas y polígonos.

Su importancia radica en que integra el análisis tabular con el análisis territorial, permitiendo realizar operaciones espaciales avanzadas de manera intuitiva.

## 2. Estructura Fundamental: GeoDataFrame

Un **GeoDataFrame** es una extensión del DataFrame tradicional.

Mientras un DataFrame clásico contiene columnas numéricas o categóricas, un GeoDataFrame añade una columna especial llamada:

`geometry`

Esta columna almacena objetos geométricos reales.

### 2.1. Tipos de geometrías

- Point
- LineString
- Polygon
- MultiPolygon

Ejemplo de creación:

```
import geopandas as gpd
from shapely.geometry import Point

data = {'ciudad': ['A', 'B'],
        'geometry': [Point(-74.1, 4.6), Point(-75.5, 6.2)]}

gdf = gpd.GeoDataFrame(data, crs="EPSG:4326")
```

## 3. Arquitectura Interna

GeoPandas funciona como un orquestador de múltiples librerías especializadas:

### 3.1. Shapely

Gestiona la representación geométrica y las operaciones espaciales:

- Intersección
- Unión
- Diferencia
- Buffer

## 3.2. PyProj

Gestiona los sistemas de referencia espacial (CRS) y transformaciones de coordenadas.

## 3.3. Fiona

Permite la lectura y escritura de formatos geoespaciales como:

- Shapefile
- GeoJSON
- GeoPackage

# 4. Sistema de Referencia Espacial (CRS)

Todo GeoDataFrame tiene un CRS asociado. Este define cómo se interpretan las coordenadas en el espacio.

Ejemplo de transformación:

```
gdf = gdf.to_crs(epsg=3857)
```

La transformación es fundamental cuando se calculan distancias, áreas o buffers.

# 5. Operaciones Espaciales

GeoPandas incorpora operaciones espaciales que no existen en pandas tradicional.

## 5.1. Buffer

```
gdf_buffer = gdf.buffer(1000)
```

Genera una zona de influencia alrededor de una geometría.

## 5.2. Spatial Join

```
gdf_resultado = gpd.sjoin(gdf1, gdf2, how="inner", predicate="intersects")
```

Permite combinar datasets basándose en relaciones espaciales.

## 5.3. Overlay

```
gdf_diff = gpd.overlay(gdf1, gdf2, how="difference")
```

Permite realizar operaciones topológicas entre capas.

## 6. Modelo Conceptual

GeoPandas puede entenderse como:

Datos tabulares + Geometría + Sistema de coordenadas

Matemáticamente, cada fila puede representarse como:

$$x_i = (atributos_i, geometría_i)$$

Donde la geometría es un objeto definido en un espacio métrico.

## 7. Aplicaciones en Ciencia de Datos

GeoPandas conecta el análisis estadístico con el análisis territorial:

- Planeación urbana
- Optimización logística
- Análisis de cobertura
- Marketing territorial
- Generación de microzonas

Permite transformar datos crudos en información espacialmente contextualizada.

## 8. Conclusión

GeoPandas no es simplemente una herramienta para visualizar mapas. Es una extensión del paradigma de análisis de datos hacia el espacio geográfico.

Permite integrar estadística, geometría y análisis territorial dentro del ecosistema de Python.

Repositorio de ejemplo: <https://github.com/MLDante/Ejemplo-GeoPandas.git>