

Clasificación de Trámites de Construcción en La Paz

...

Proyecto Adquisición y
Comprensión de datos

Integrantes:

- Ericka Cori
- Paolo Ramos
- Gaston Nina

Introducción

La gestión de trámites de construcción es clave para un desarrollo urbano ordenado. Este estudio aplica técnicas de Adquisición y comprensión de datos sobre datos municipales de La Paz, con el objetivo de predecir si un trámite será aprobado u observado. A través de algoritmos de clasificación, se analizan características técnicas, territoriales y legales para apoyar una mejor toma de decisiones administrativas.



Objetivo del Proyecto

Dar una alerta temprana si un trámite municipal de construcción en La Paz será aprobado u observado, utilizando atributos técnicos, territoriales y legales disponibles antes de su revisión oficial.



Dataset utilizado

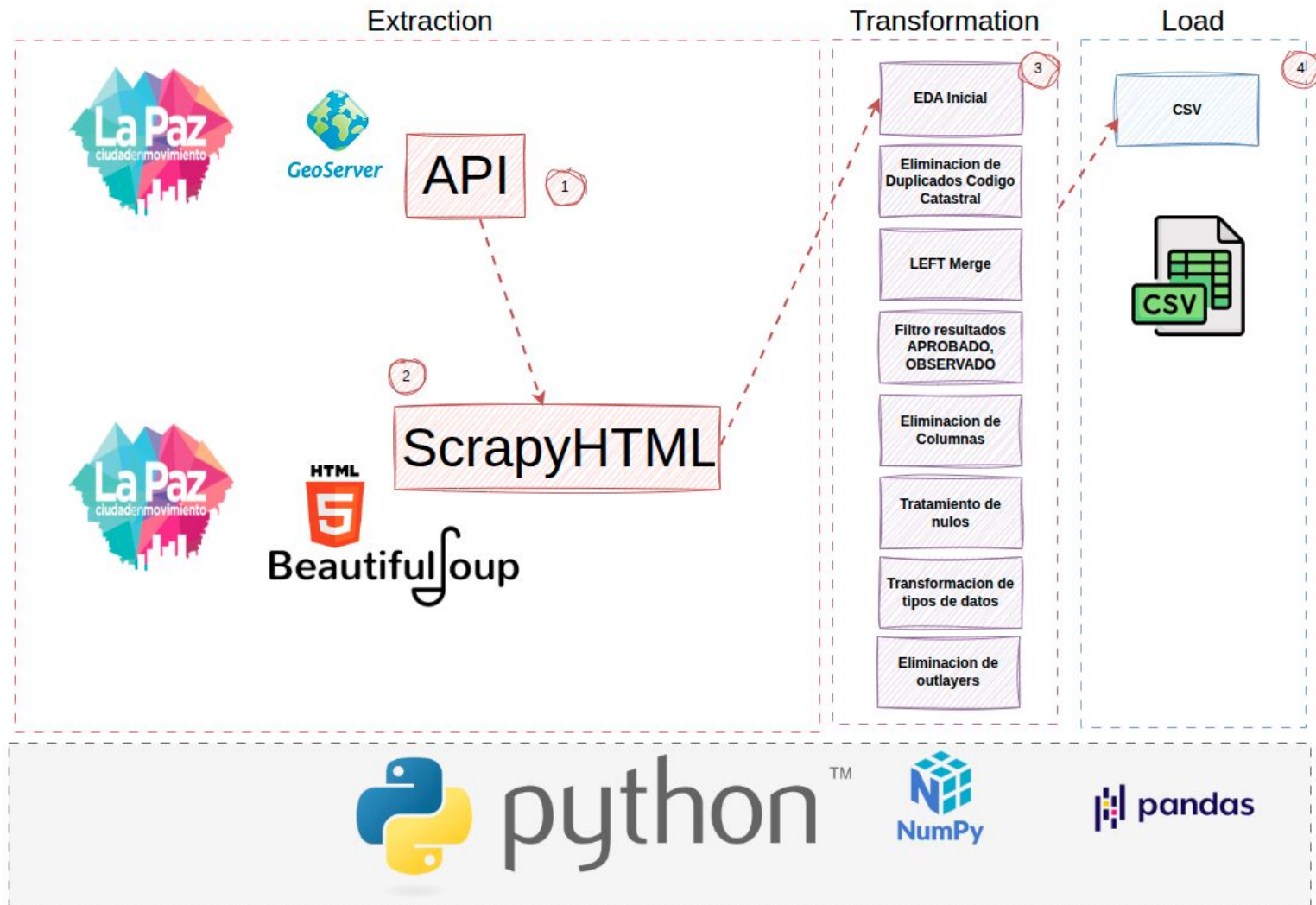
El proceso metodológico se estructuró según las fases del ciclo de vida de un proyecto de análisis de datos. En la fase de Generación y Captura de Datos, se utilizó un dataset público de trámites municipales de construcción en La Paz, compuesto por más de 13,000 registros.

Las variables clave incluyen:

- Tipo de proyecto
- Cantidad de pisos
- Superficie legal
- Distrito municipal
- Resultado del trámite (Aprobado / Observado)



DIAGRAMA DE FLUJO



EXTRACTION


Fuente de Datos 1 - API - GEOSERVER

sit.servicios.lapaz.bo/mapas/territorio/permisos-construccion.html

PERMISOS DE CONSTRUCCIÓN

Permisos Aprobados Mapas

Gestión para ver los trámites de Permisos de Construcción solicitados en el Municipio de La Paz. Gestión 2025 Se aprobaron 90 permisos de construcción



Permisos de Construcción

Fecha Ingreso	Macrodistrito	Código Catastral	Profesional Responsable	Pisos Fachada
2025-04-03	III - PERIFERICA	006013500060000	RONEY HUGO MACEDA COR...	4
2025-02-22	III - PERIFERICA	006162800060000	ROLANDO JOAQUIN BARRE...	3
2025-02-17	III - PERIFERICA	006007400080000	ARIEL HUANCA RODRIGUEZ	5
2025-02-26	III - PERIFERICA	036024600050000	ROBERTO OMAR MACUCHA...	2
2025-02-28	III - PERIFERICA	006066800120000	BLANCA MARIA QUISPE LIMA...	4
2025-01-10	IV - SAN ANTONIO	020026600080000	VICTOR ALBERTO SARAVIA ...	4
2024-12-30	IV - SAN ANTONIO	036005500270000	PAOLA VERÓNICA PHILCO P...	8

Sistema de Información Territorial Secretaría Municipal de Planificación

Revisando la plataforma es un sitio web que consulta datos a un geoserver filtrando información solo Aprobada y de año específico

Web:

<https://sit.servicios.lapaz.bo/mapas/territorio/permisos-construccion.html>

Captura de Datos

```
{
  type : FeatureCollection
  totalFeatures : 13061
  Features : [ 15000 Items
    0 : {
      type : Feature
      id : tramitesterritoriales.1
      geometry : {
        type : Point
        coordinates : [ 2 Items
          0 : -68.1085476
          1 : -16.5207948
        ]
      }
      geometry_name : geom
      properties : {
        idPCTramite : 1
        descripcion : PERMISO DE CONSTRUCCION
        idTipoTramite : 1
        fechaRegistro : 2017-11-14T17:04:40.303Z
        Solicitante : SALVADOR MAURICIO REVOLLO ALARCON
        arquitectoNombre : EDSON SANJINEZ RAMOS
        arquitectoRegistroNacionalCAB : 5047
        nroInmueble : 60159
        idProyectoDesarrollo : 8
        idTipoObra : 2
        TipoProyecto : Residencial
        TipoObra : Construcción Nueva
        NumeroTramite : 1
        Resultado : OBSERVADO
        EstadoTramite : FINALIZADO
        idInsDocumento : null
        nombreArchivo : null
        FechaRegistroArch : null
        codigoCatastral : 017009100040000
        fechaAprobacion : 2017-12-12T20:22:07.723Z
        macroDistrito : V - SUR
        distritoMunicipal : 21
        cantidadPisos : 4
        superficielegal : null
        nombreEdificio : VIVIENDAS COPLAT
        SuperficieConstruida : null
      }
    }
  ]
}
```

Revisando el URL podemos ver:

1. Que la data esta filtrada, por campos especificos
2. Tiene un where de: **Resultado=APROBADO** y **FechaAprobacion >= 2025-01-01 and 2025-12-31**

También el objeto JSON obtenido con los campos

Web: LINK

GEThttps://sit.servicios.lapaz.bo/geoserver/sit/ows?service=WFS&version=1.0.0&request=GetFeature&typeName=sit:tramitesterritoriales&propertyName=idPCTramite

Params

Authorization

Headers (7)

Body

Scripts

Settings

<input checked="" type="checkbox"/>	service	WFS
<input checked="" type="checkbox"/>	version	1.0.0
<input checked="" type="checkbox"/>	request	GetFeature
<input checked="" type="checkbox"/>	typeName	sit:tramitesterritoriales
<input checked="" type="checkbox"/>	propertyName	idPCTramite,idTipoTramite,descripcion,fechaRegistro,,arquitectoNombre,idTL...
<input checked="" type="checkbox"/>	maxFeatures	1000
<input checked="" type="checkbox"/>	outputFormat	application%2Fjson
<input checked="" type="checkbox"/>	srsName	EPSG:4326
<input checked="" type="checkbox"/>	CQL_FILTER	Resultado%20%3D%20%27APROBADO%27%20AND%20fechaAprobacion%2...

CAPTURA DE DATOS

```
# Parámetros base
base_url = "https://sit.servicios.lapaz.bo/geoserver/sit/ows"
params = {
    "service": "WFS",
    "version": "2.0.0",
    "request": "GetFeature",
    "typeName": "sit:tramites_territoriales",
    "count": 5000,
    "outputFormat": "application/json",
    "srsName": "EPSG:4326"
}

# Número total de features (puedes ajustar si cambia)
total_features = 13061
features_per_page = params["count"]
paginas = range(0, total_features, features_per_page)

# Lista para guardar los registros
todos_los_registros = []

# Descarga paginada
for start_index in tqdm(paginas, desc="Descargando páginas"):
    params["startIndex"] = start_index
    response = requests.get(base_url, params=params)

    if response.status_code == 200:
        data = response.json()
        for feature in data["features"]:
            props = feature["properties"]
            geometry = feature.get("geometry")

            # Validar que la geometría exista
            if geometry and "coordinates" in geometry:
                lon, lat = geometry["coordinates"]
                props["latitude"] = lat
                props["longitude"] = lon
            else:
                props["latitude"] = None
                props["longitude"] = None

            todos_los_registros.append(props)
    else:
        print(f"⚠ Error en startIndex={start_index}: código {response.status_code}")

# Crear DataFrame
df = pd.DataFrame(todos_los_registros)

# Guardar como CSV
df.to_csv("tramites_lapaz_completo.csv", index=False, encoding="utf-8")

# Mostrar primeras filas
df.head()
```

Código Python para extracción

Realizamos un script para poder recorrer todas las páginas de la data y guardarla en un CSV para mejor manipulación

LINK Codelab

<https://colab.research.google.com/drive/1splkFkcozaYoIh0UFaXyyMbckndXln8E>

Fuente de Datos 2 - Scrappy - HTML

sitservicios.lapaz.bo/situtiles/pc/?MDQ0MTA0OTAwMDcwMDAwfDYxMTUzfDIwMjU=

Gobierno Autónomo Municipal de La Paz • Secretaría Municipal de Planificación

PERMISO DE CONSTRUCCION-LICENCIA AGIL N°

61153/2025

Datos Generales

Código Catastral:	044 1049 0007 0000
Informe:	DDAT PSAT 960/2025
Fecha Aprobación:	21/mar./2025
Zona Referencial:	IRPAVI II
Patrón de Asentamiento:	4P - d18

Parámetros de Construcción

Sur 4P - d18		Parámetros Autorizados	
ALE	Área de Lote Edificable		516.80 m2
FML	Frente Mínimo de Lote		48.84 ml
AMC	Área Máxima a Cubrir	Sótano	0.00 m2
		Semisótano	0.00 m2
		Zócalo	0.00 m2
		Mezzanine	0.00 m2
		Torre	127.83 m2
AME	Área Máxima a Edificar	Zócalo	0.00 m2
		Torre	354.04 m2
AMF	Altura Máxima de Fachada	Sótano	0 plantas
		Semisótano	0 plantas

Sistema de Información Territorial Secretaría Municipal de Planificación

Revisando la plataforma es un sitio web que consulta datos a un geoserver filtrando información solo Aprobada y de año específico

Web:

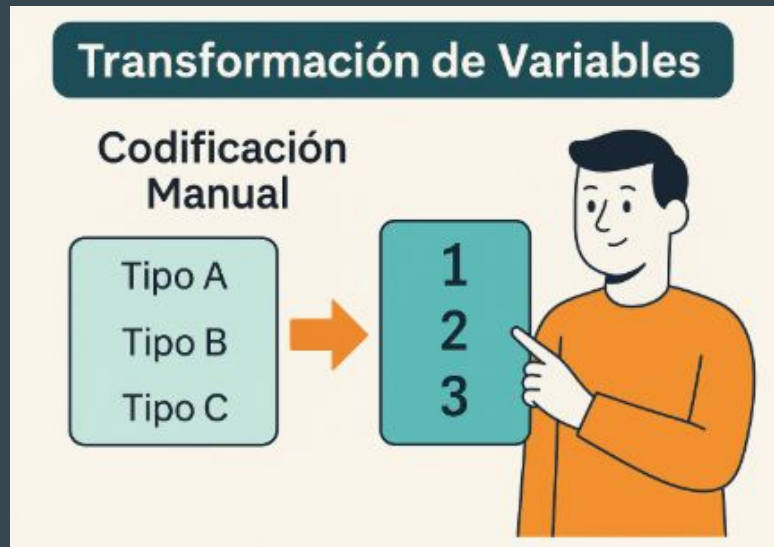
<https://sitservicios.lapaz.bo/situtiles/pc/?MDQ0MTA0OTAwMDcwMDAwfDYxMTUzfDIwMjU=>

TRANSFORMATION

Transformación de Datos

Las variables categóricas fueron preparadas mediante codificación manual, convirtiendo cada categoría en su representación numérica correspondiente.

Esta transformación es esencial para que los algoritmos de aprendizaje automático puedan interpretar y procesar correctamente los datos durante el entrenamiento del modelo.



EDA Inicial

Analizamos la estructura, calidad y relaciones iniciales del dataset de trámites catastrales.

Hallazgos de la data

- Volumen: 17,907 registros iniciales con 31 columnas
- Composición de datos:
 - 9 columnas numéricas (float64)
 - 5 columnas enteras (int64)
 - 17 columnas de texto (object)

Problemas encontrados

- id_ins_documento: 87.5% nulos (1,735/13,907)
- nombre_archivo: 87.5% nulos (1,735/13,907)
- superficie_construida: 73.6% nulos (3,675/13,907)
- nombre_edificio: 83.8% nulos (2,259/13,907)

Eliminación de duplicados de duplicados Código Catastral

- Análisis:
 - 3,510 códigos catastrales duplicados
 - Algunos códigos se repiten hasta 22 veces
 - Data inicial de aproximadamente 17,000 registros
- Solución aplicada:
 - Agrupación por codigo_catastral
 - LEFT JOIN para mantener el registro más representativo
 - Eliminación de duplicados conservando información relevante

LEFT Merge

Se realizó un margen entre nuestras dos fuentes

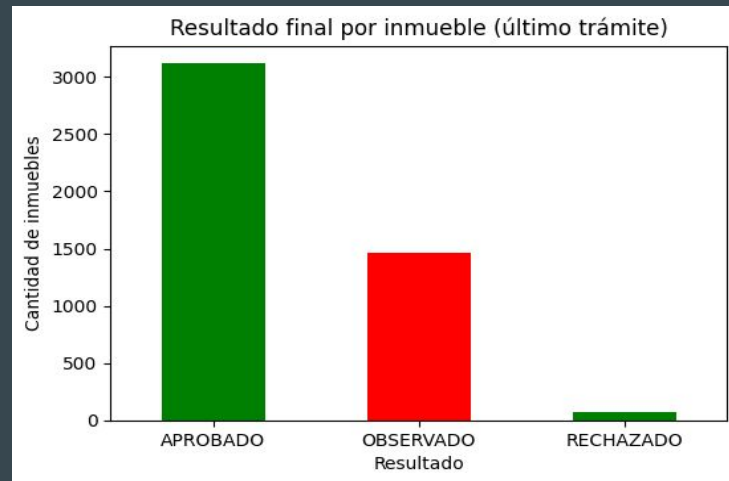
```
df_html = pd.read_csv("tramites_html_identificador.csv", dtype={"gral_codigo_catastral": str})

# Hacemos left merge desde df_sin_duplicados hacia df_html
df_merge = pd.merge(
    df_filtrado,      # la base principal
    df_html,          # datos adicionales
    on="identificador_b64", # campo común
    how="left"
)

# Verificar resultado
print(df_merge.shape)
df_merge.head()
```

Filtro resultados APROBADO y OBSERVADO

- 100% trámites con estado "Aprobado"
- Eliminación de trámites rechazados, pendientes y en revisión
- Base consistente para análisis de patrones de aprobación



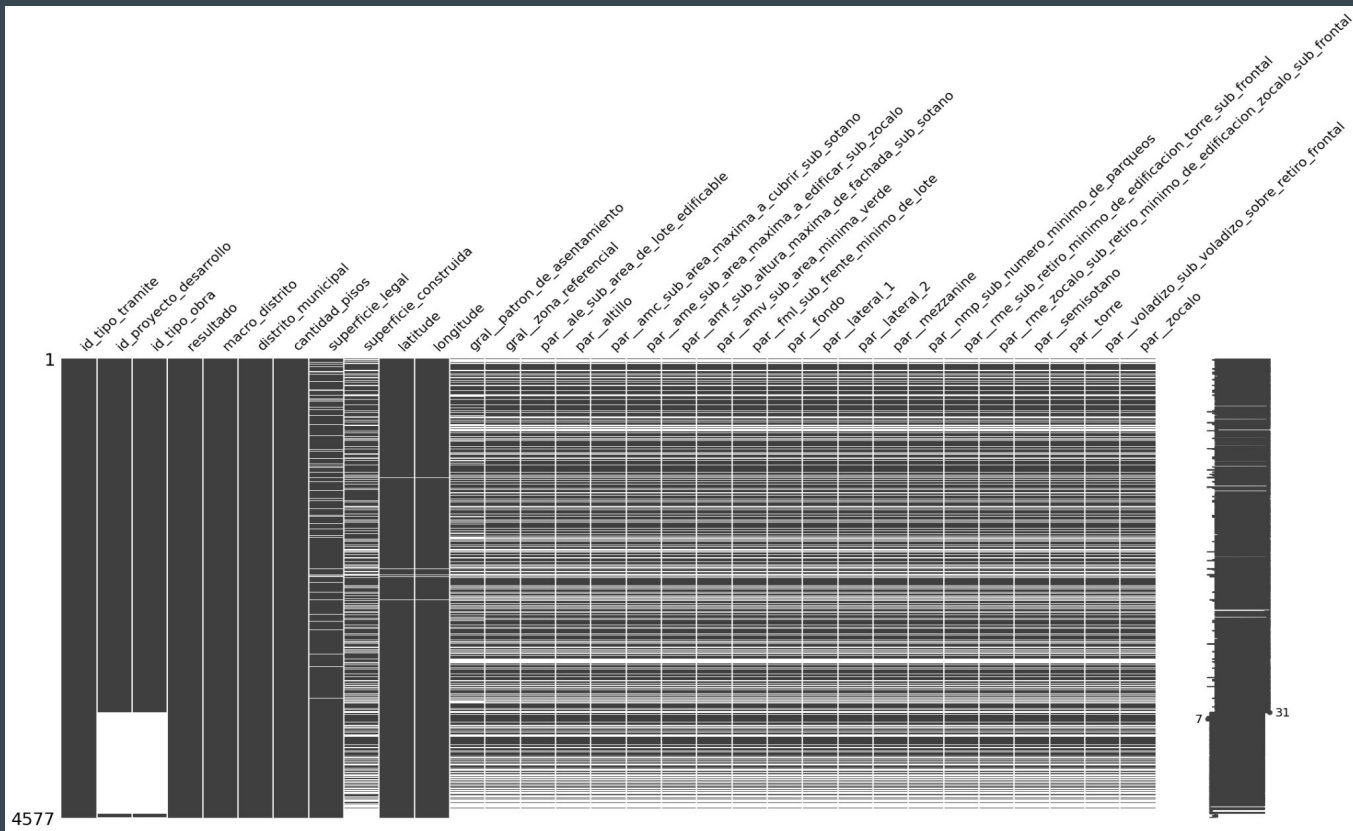
Eliminación de Columnas

Impacto en el Dataset

- Antes: 31 columnas
- Después: ~18 columnas (42% reducción dimensional)
- Beneficios: Mejor performance y enfoque analítico

```
# Columnas que no son relevantes para el análisis supervisado
cols_drop = [
    "id_pc_tramite", "descripcion", "numero_tramite",
    "identificador", "identificador_b64",
    "codigo_catastral",
    "fecha_registro", "fecha_aprobacion",
    "gral_codigo_catastral", "gral_fecha_aprobacion",
    "nombre_archivo", "url", "error",
    "nombre_edificio",
    "solicitante", "arquitecto_nombre", "otro_arquitecto_reponsable",
    "arquitecto_registro_nacional_cab",
    "id_ins_documento",
    "fecha_registro_arch",
    "nro_inmueble",
    "estado_tramite",
    "tipo_proyecto",
    "tipo_obra",
    "fecha_registro_arch",
    "codigo_catastral",
    "anio_registro",
    "gral_informe",
    "gral_normativa_segun_informe",
    "otro_tipo_de_obra",
    "otro_tipo_de_proyecto",
]
```

Tratamiento de nulos



1. IDs Internos Sin

Valor Analítico

2. Columnas con Alto

Porcentaje de Nulos

(>90%)

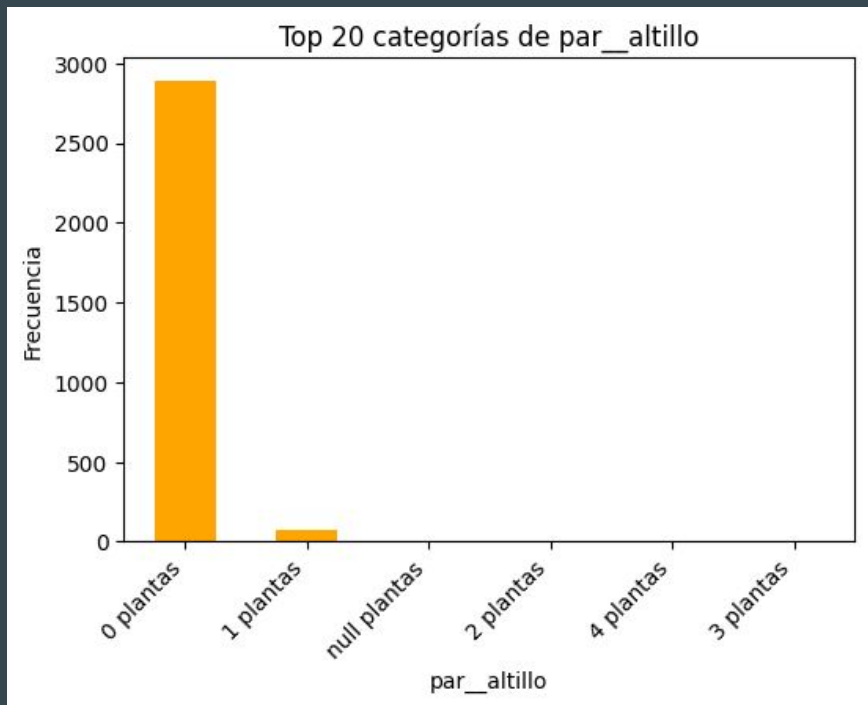
3. Información

Redundante o

Técnica

Transformación de tipos de datos

- Se realizó normalización temporal
- Transformación de columnas categóricas



```
# id_tipo_tramite - convertimos es_agil
df_modelo["es_agil"] = df_modelo["id_tipo_tramite"] == 12

# id_proyecto_desarrollo
# Reemplazar NaN por 0
df_modelo["id_proyecto_desarrollo"] = df_modelo["id_proyecto_desarrollo"].fillna(12)
# Convertir a integer (int)
df_modelo["id_proyecto_desarrollo"] = df_modelo["id_proyecto_desarrollo"].astype(int)

# convertimos tipo de obra en es ampliacion?
# Reemplazar NaN por 1 primero 1= Construccion Nueva
df_modelo["id_tipo_obra"] = df_modelo["id_tipo_obra"].fillna(1).astype(int)
# Crear campo booleano: True = Ampliación, False = Nueva
df_modelo["es_ampliacion"] = df_modelo["id_tipo_obra"] == 2

# Cantidad de pisos
# Reemplazar NaN por 0
df_modelo["cantidad_pisos"] = df_modelo["cantidad_pisos"].fillna(0)
# Convertir a entero (int)
df_modelo["cantidad_pisos"] = df_modelo["cantidad_pisos"].astype(int)

# Borramos los que son superficie_legal Nan
df_modelo = df_modelo.dropna(subset=["superficie_legal"])

# Superficie construida
# Reemplazar NaN por 0
df_modelo["superficie_construida"] = df_modelo["superficie_construida"].fillna(0)

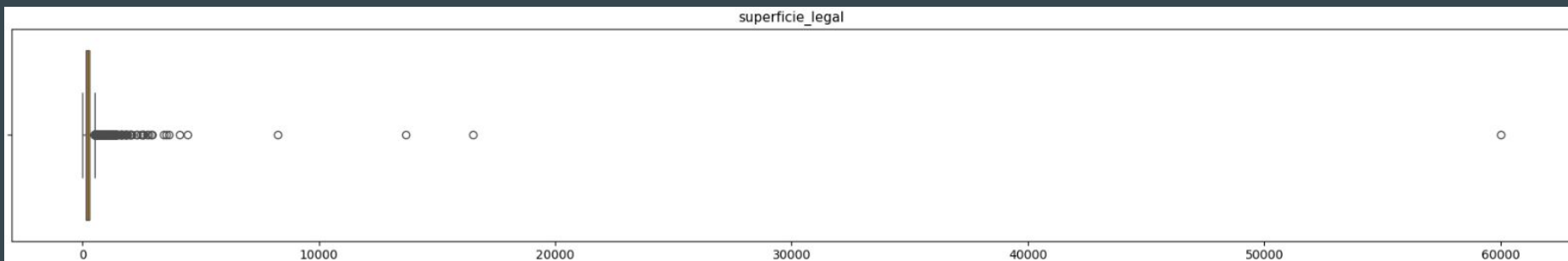
# reducimos lon/lat a 6 decimales para no hacerlos tan especificos
df_modelo["latitude"] = df_modelo["latitude"].round(6)
df_modelo["longitude"] = df_modelo["longitude"].round(6)
```


Eliminación de outlayers

- Se detectaron columnas con outliers: En la columna de pisos, superficie legal, pared de fondo entre otros

Filas con superficie_legal > 5000: 4

	superficie_legal	superficie_construida	resultado	macro_distrito	id_proyecto_desarrollo
2311	60000.00	29282.77	APROBADO	VII - CENTRO	10
4029	16533.94	32443.74	APROBADO	I - COTAHUMA	12
793	13687.12	0.00	OBSERVADO	V - SUR	4
4247	8257.59	0.00	OBSERVADO	V - SUR	12



A black and white photograph of a city, likely Lima, Peru, featuring a large stadium (Estadio Nacional) and a snow-capped mountain (Huascarán) in the background. The word "LOAD" is overlaid in large white letters.

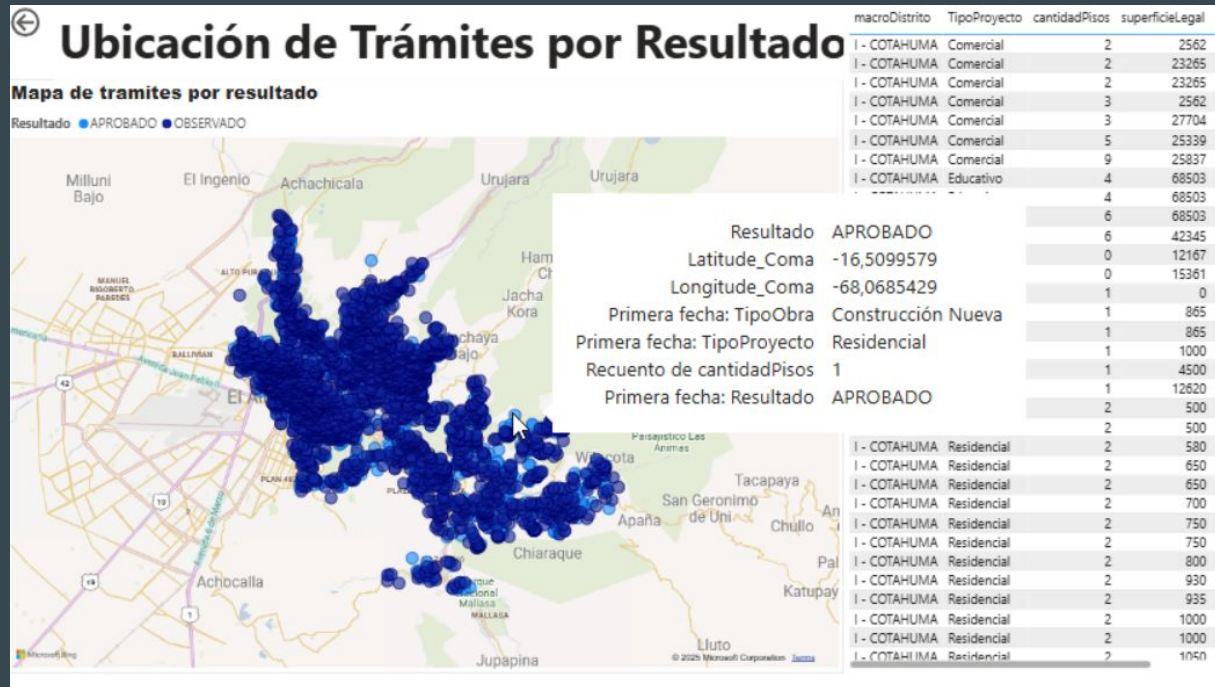
LOAD

Load

```
df_modelo.to_csv("df_modelo_limpio_1.csv",index=False)
```

Cargar el dataset transformado
en un almacén de datos.

En este caso generamos un
archivo csv, para que pueda ser
utilizado desde un Tableau,
Power BI, Excel Reprts entre
otros.



A black and white aerial photograph of a city, likely La Paz, Bolivia. The city is built on a steep hillside, with a dense concentration of buildings in the foreground and middle ground. In the background, a large, snow-capped mountain (Mount Illimani) rises above the city. The sky is cloudy. The word "RESULTADOS" is overlaid in large, white, sans-serif capital letters across the center of the image.

RESULTADOS

Columnas finales

[illegible]

Columnas finales

```
df_modelo.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
Index: 4305 entries, 0 to 4576
```

```
Data columns (total 33 columns):
```

#	Column	Non-Null Count	Dtype
0	id_tipo_tramite	4305 non-null	int64
1	id_proyecto_desarrollo	4305 non-null	int64
2	id_tipo_obra	4305 non-null	int64
3	resultado	4305 non-null	category
4	macro_distrito	4305 non-null	category
5	distrito_municipal	4305 non-null	category
6	cantidad_pisos	4305 non-null	int64
7	superficie_legal	4305 non-null	float64
8	superficie_construida	4305 non-null	float64
9	latitude	4305 non-null	float64
10	longitude	4305 non-null	float64
11	gral_patron_de_asentamiento	4305 non-null	category
12	gral_zona_referencial	4305 non-null	category
13	par_ale_sub_area_de_lote_edificable	4305 non-null	float64
14	par_attillo	4305 non-null	int64
15	par_amc_sub_area_maxima_a_cubrir_sub_sotano	4305 non-null	float64
16	par_ame_sub_area_maxima_a_edificar_sub_zocalo	4305 non-null	float64
17	par_amf_sub_altura_maxima_de_fachada_sub_sotano	4305 non-null	int64
18	par_amv_sub_area_minima_verde	4305 non-null	float64
19	par_fml_sub_frente_minimo_de_lote	4305 non-null	float64
20	par_fondo	4305 non-null	float64
21	par_lateral_1	4305 non-null	float64
22	par_lateral_2	4305 non-null	float64
23	par_mezzanine	4305 non-null	float64
24	par_nmp_sub_numero_minimo_de_parqueos	4305 non-null	int64
25	par_rme_sub_retiro_minimo_de_edificacion_torre_sub_frontal	4305 non-null	float64
26	par_rme_zocalo_sub_retiro_minimo_de_edificacion_zocalo_sub_frontal	4305 non-null	float64
27	par_semisotano	4305 non-null	int64
28	par_torre	4305 non-null	int64
29	par_voladizo_sub_voladizo_sobre_retiro_frontal	4305 non-null	float64
30	par_zocalo	4305 non-null	int64
31	es_agil	4305 non-null	bool
32	es_ampliacion	4305 non-null	bool

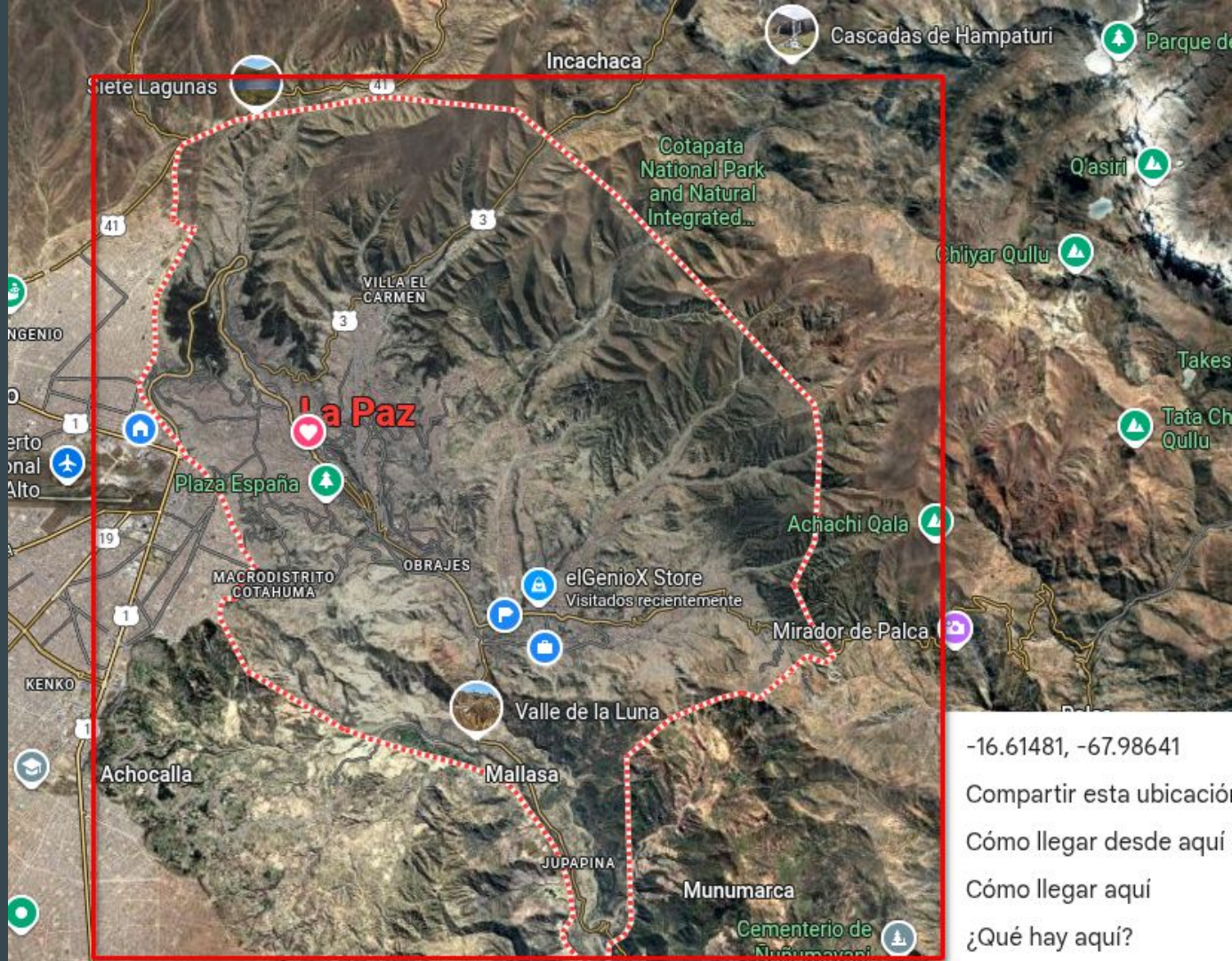
```
dtypes: bool(2), category(5), float64(16), int64(10)
```

```
memory usage: 953.3 KB
```

GREAT EXPECTATIONS

Rangos Geográficos
de La Paz

para asegurar no se
registra una
ubicación fuera del
municipio.



-16.61481, -67.98641

Compartir esta ubicación

Cómo llegar desde aquí

Cómo llegar aquí

¿Qué hay aquí?

GREAT EXPECTATIONS

Validar y Generar Reporte

Ejecutar validación

```
validation_result = ge_df_with_expectations.validate()
```

Mostrar resultados

```
print("=== RESULTADOS DE VALIDACIÓN ===")
print(f"Validación exitosa: {validation_result.success}")
print(f"Total expectativas: {len(validation_result.results)}")
```

Detalles de fallos

```
failed_expectations = [
    result for result in validation_result.results
    if not result.success
]
```

```
print(f"Expectativas fallidas: {len(failed_expectations)}")
```

for failed in failed_expectations:

```
    print(f"❌ {failed.expectation_config.expectation_type}")
    print(f"    Columna: {failed.expectation_config.kwargs.get('column', 'N/A')}")
    print(f"    Detalle: {failed.result}")
```

```
=== RESULTADOS DE VALIDACIÓN ===
```

```
Validación exitosa: True
```

```
Total expectativas: 10
```

```
Expectativas fallidas: 0
```

```
def create_data_expectations(ge_df):
    # Expectativa para latitud (La Paz aprox: -16.7 a -16.4)
    ge_df.expect_column_values_to_be_between(
        column="latitude",
        min_value=-16.61448,
        max_value=-16.43148,
    )

    # Expectativa para longitud (La Paz aprox: -68.2 a -68.0)
    ge_df.expect_column_values_to_be_between(
        column="longitude",
        min_value=-68.17695,
        max_value=-68.00735,
    )

    # cantidad de pisos <= 40
    ge_df.expect_column_values_to_be_between(
        "cantidad_pisos",
        min_value=0,
        max_value=40
    )

    # Superficies no negativas
    ge_df.expect_column_values_to_be_between(
        "superficie_legal",
        min_value=0, # porque puede ser razon para que falle como OBSERVADO
        max_value=5000
    )
    ge_df.expect_column_values_to_be_between(
        "superficie_construida",
        min_value=0,
        max_value=35000
    )

    # resultado
    ge_df.expect_column_values_to_be_in_set(
        column="resultado",
        value_set=["APROBADO", "OBSERVADO"]
    )

    # sin nulos
    ge_df.expect_column_values_to_not_be_null("macro_distrito")
    ge_df.expect_column_values_to_not_be_null("distrito_municipal")

    # Si tienes variables booleanas (es_agil, es_ampliacion)
    ge_df.expect_column_values_to_be_in_set("es_agil", [True, False])
    ge_df.expect_column_values_to_be_in_set("es_ampliacion", [True, False])
```


AIRFLOW

eda_inicial	■	■	■
eliminar_duplicados	■	■	■
left_merge	■	■	■
filtrar_resultados	■	■	■
eliminar_columnas	■	■	■
tratar_nulos	■	■	■
transformar_tipos	■	■	■
eliminar_outliers	■	■	■

DAG **tramites_lapaz_transform** Run 2025-10-04, 02:04:11 -04 / Task **eliminar_dupl**

[Details](#) [Graph](#) [Gantt](#) [Code](#) [Event Log](#) [Logs](#)



A black and white aerial photograph of a densely populated city, likely La Paz, Bolivia. The city is built on a steep hillside, with numerous high-rise apartment buildings and smaller residential structures. In the background, a large, snow-capped mountain (Mount Illimani) rises above the city, partially obscured by clouds. The sky is filled with soft, white clouds. The word "GRACIAS!" is superimposed in large, white, sans-serif capital letters across the center of the image.

GRACIAS!

A black and white aerial photograph of a city, likely La Paz, Bolivia. The city is built on a steep hillside, with a dense concentration of buildings in the foreground and middle ground. In the background, a large, snow-capped mountain (Mount Illimani) rises above the city. The sky is cloudy. The word "CÓDIGO" is overlaid in large, white, sans-serif capital letters across the center of the image.

CÓDIGO