



# Proyecto Final Sistema de Gestión Inmobiliaria

# Problemática

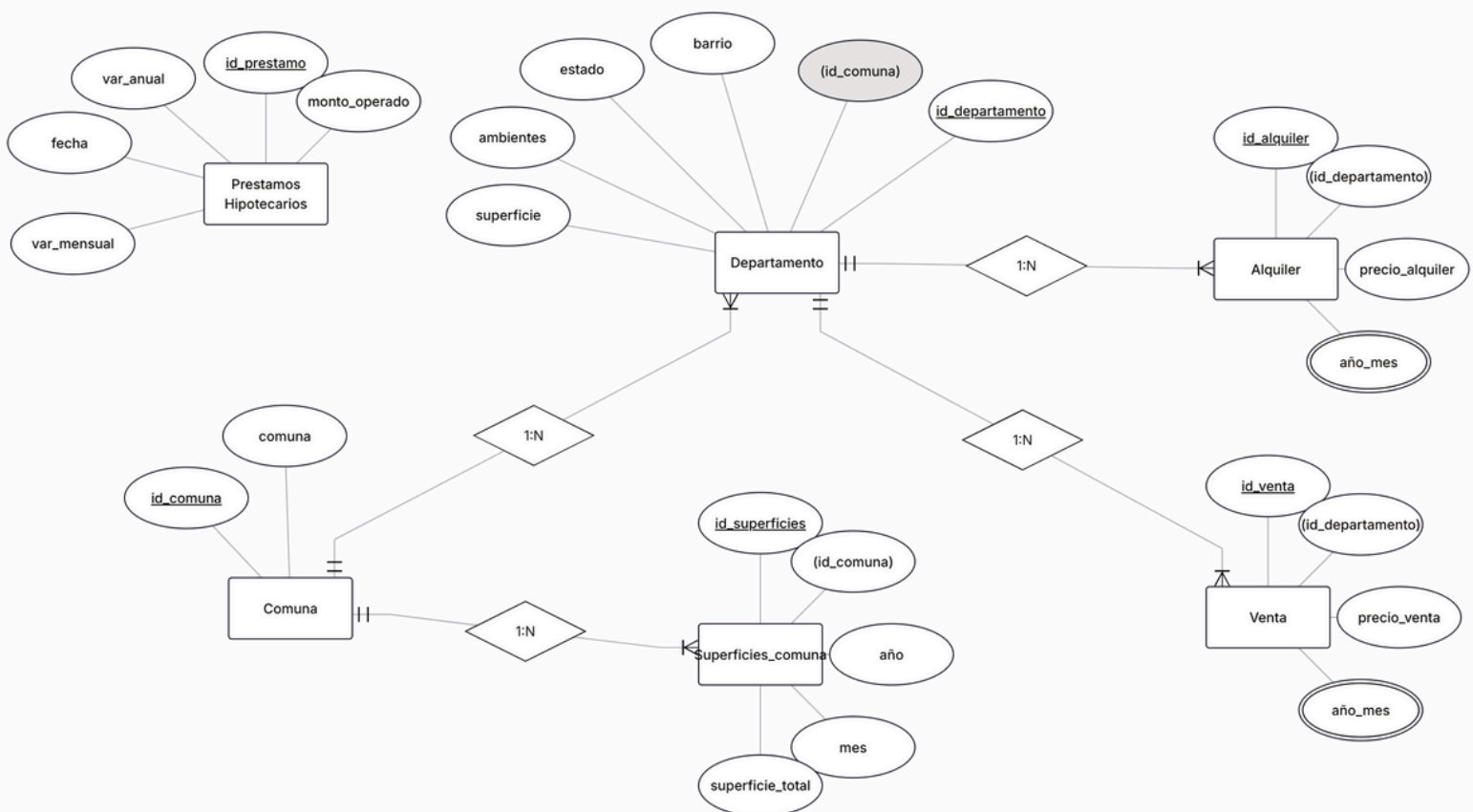
En el mercado inmobiliario, luego de la derogación de la ley de alquileres, los valores de las propiedades registraron una baja cercana al 30%. Sin embargo, acceder a la primera vivienda sigue siendo un desafío: requiere años de ahorro y un gran esfuerzo económico.

Con este proyecto armé una base de datos que busca responder preguntas clave: ¿qué zonas de la ciudad ofrecen mayor accesibilidad y rentabilidad?, ¿cómo evolucionan los precios con el tiempo?, ¿qué relación existe entre los créditos hipotecarios otorgados y la compra de viviendas?

De esta forma, el análisis pretende aportar una mirada más completa sobre el impacto real que tienen las regulaciones y el contexto económico en la dinámica habitacional de la Ciudad de Buenos Aires.

Cabe aclarar que inicialmente consideré incluir las declaraciones patrimoniales de funcionarios públicos, pero decidí no hacerlo ya que ese dataset está más orientado al análisis de transparencia del sector público que al mercado inmobiliario en general. Incluirlo hubiera desviado el foco del proyecto, que se centra en la evolución de precios, créditos y accesibilidad habitacional.

## Diagrama Entidad - Relación



En este diagrama, puse en subrayado los campos que representan una PK y los que están entre paréntesis los que representan una FK.

# Listado de tablas

## Tabla Departamentos

Campo	Tipo de Dato	PK	FK	NULL
id_departamento	INT	X		
barrio	VARCHAR			
superficie	DECIMAL			
ambientes	INT			
estado	VARCHAR			
id_comuna	INT		X	

## Tabla Alquiler

Campo	Tipo de dato	PK	FK	NULL
id_alquiler	INT	X		
id_departamentos	INT		X	
precio_alquiler	DECIMAL			
año_mes	DATE			

## Tabla Venta

Campo	Tipo de dato	PK	FK	NULL
id_departamento	INT		X	
id_venta	INT	X		
precio_venta	DECIMAL			
año_mes	DATE			

## Tabla Prestamos Hipotecarios

Campo	Tipo de dato	PK	FK	NULL
id_prestamo	INT	X		
monto_operado	DECIMAL			
var_mensual	DECIMAL			
var_anual	DECIMAL			
año_mes	DATE			

## Tabla Superficies Comuna

Campo	Tipo de dato	PK	FK	NULL
id_superficie	INT	X		
id_comuna	INT		X	
superficie_total	DECIMAL			
año_mes	DATE			

## Tabla Comuna

Campo	Tipo de dato	PK	FK	NULL
id_comuna	INT	X		
comuna	VARCHAR			

# Listado de objetos creados

## Vistas

Descripción	Objetivo	Tablas
Vista de Precios Promedio por Comuna	Ver qué comunas son más caras para comprar/alquilar.	Comuna, Departamentos, Venta, Alquiler.
Vista de Departamentos Más Rentables (Alquiler)	Identificar qué propiedades tienen mejor retorno de inversión.	Departamentos, Alquiler, Venta

## Funciones

### 1. calcular\_precios\_m2

Descripción	Objetivo	Tablas que manipula
Calcula el precio por metro cuadrado de un departamento específico basándose en su último precio de venta y superficie.	Permitir comparar el valor de diferentes propiedades de manera estandarizada, facilitando análisis de mercado.	<ul style="list-style-type: none"><li>• <b>Venta (lectura):</b> Obtiene el precio de venta más reciente</li><li>• <b>Departamentos (lectura):</b> Obtiene la superficie del departamento</li></ul>

## 2. categorizar\_precio

Descripción	Objetivo	Tablas que manipula
Clasifica un precio de venta en categorías (Económico, Medio, Alto, Premium) según rangos predefinidos.	Facilitar la segmentación del mercado inmobiliario y permitir análisis por categoría de precio.	<ul style="list-style-type: none"><li>• <b>Venta (lectura):</b> Obtiene el precio de venta más reciente</li><li>• <b>Departamentos (lectura):</b> Obtiene la superficie del departamento</li></ul>

## Stored Procedures:

### 1.registrar\_venta

Descripción	Objetivo	Tablas que manipula	Parámetros
Registra una nueva operación de venta con validación previa de existencia del departamento.	Automatizar el proceso de registro de ventas asegurando la integridad referencial y proporcionando feedback al usuario.	<ul style="list-style-type: none"><li>• <b>Departamentos (lectura):</b> Valida existencia del departamento</li><li>• <b>Venta (escritura):</b> Inserta el nuevo registro</li></ul>	<b>p_id_departamento (INT):</b> ID del departamento vendido <b>p_precio (DECIMAL(12,2))</b> : Precio de venta <b>p_fecha (DATE):</b> Fecha de la operación

## 2. reporte\_mercado\_comuna

Descripción	Objetivo	Tablas que manipula	Parámetros
Genera un reporte completo de estadísticas del mercado inmobiliario para una comuna específica.	Proporcionar análisis agregado de precios, cantidades y características de departamentos por comuna, facilitando la toma de decisiones de inversión.	<ul style="list-style-type: none"><li>• <b>Comuna (lectura):</b> Obtiene nombre de la comuna</li><li>• <b>Departamentos (lectura):</b> Cuenta y promedia características</li><li>• <b>Venta (lectura):</b> Calcula estadísticas de precios de venta</li><li>• <b>Alquiler (lectura):</b> Calcula estadísticas de precios de alquiler</li></ul>	<b>p_id_comuna (INT):</b> ID de la comuna a analizar

## Triggers:

### 1.validar\_datos\_departamentos

Descripción	Cuándo se activa	Sobre qué tabla	Para qué sirve
Valida la integridad de los datos antes de insertar un nuevo departamento en la base de datos.	BEFORE INSERT en la tabla Departamentos	<ul style="list-style-type: none"><li>• <b>Departamentos</b></li></ul>	<ul style="list-style-type: none"><li>• Asegurar que la superficie sea mayor a 0</li><li>• Verificar que la cantidad de ambientes sea válida (entre 1 y 10)</li><li>• Prevenir la inserción de datos inconsistentes que puedan afectar análisis posteriores</li></ul>



2. reporte\_mercado\_comuna

Descripción	Cuándo se activa	Sobre qué tabla	Para qué sirve
Registra automáticamente todos los cambios realizados en los precios de venta, manteniendo un historial completo de modificaciones.	AFTER UPDATE en la tabla Venta	<ul style="list-style-type: none"><li>• <b>Venta (con registro en Auditoria_Precios)</b></li></ul>	<ul style="list-style-type: none"><li>• Mantener trazabilidad de cambios de precios</li><li>• Identificar quién y cuándo modificó un precio</li><li>• Permitir análisis histórico de ajustes de precios</li><li>• Cumplir con requisitos de auditoría</li></ul>

# Importación de Datos

## Paso a paso para la carga de datos:

### 1. Tabla: Comuna

- Método: Asistente de Importación
- Archivo: comuna.csv
- Registros: 15
- Formato: UTF-8, separador coma
- Columnas: nombre\_comuna
- Observación: Primera tabla cargada por no tener dependencias.

### 2. Tabla: Prestamos Hipotecarios

- Método: Mockaroo + Procesamiento Python
- Archivo original: montos.txt
- Archivo procesado: prestamos\_final.csv
- Registros: 648
- Proceso:
  1. Archivo original tenía formato argentino (puntos como separadores de miles, comas como decimales)
  2. Script Python detectó codificación (UTF-16)
  3. Limpieza de números: eliminación de comillas triples, conversión de formato
  4. Conversión de fechas de "2016-05" a "2016-05-01"
  5. Generación de fechas aleatorias válidas entre 2016-05-01 y 2021-04-30
- Columnas: monto\_operado, variacion\_mensual, variacion\_anual, año\_mes

### 3. Tabla: Departamentos

- Método: Asistente de Importación
- Archivo: departamentos.csv
- Registros: 3085
- Columnas: barrio, id\_comuna, ambientes, estado, superficie
- Observación: Requiere que existan registros en tabla Comuna (FK).

#### 4. **Tabla: Alquiler**

- Método: Generación desde archivo Excel + Python
- Archivo original: Excel con columnas separadas (mes/año)
- Archivo procesado: alquiler\_final.csv
- Registros: 1826
- Proceso:
  1. Lectura de archivo con diferentes codificaciones (latin-1 detectado)
  2. Conversión de columnas "mes" (texto) y "año" (número) a formato DATE
  3. Mapeo de meses: Ene→01, Feb→02, etc.
  4. Asignación aleatoria de id\_departamento entre 1 y 3085
  5. Formato final: YYYY-MM-DD
- Columnas: id\_departamento, precio\_alquiler, año\_mes

#### 5. **Tabla: Venta**

- Método: Similar a Alquiler
- Archivo procesado: ventas\_final.csv
- Registros: 3085
- Proceso: Idéntico a tabla Alquiler
- Columnas: id\_departamento, precio\_venta, año\_mes

#### **Problemas encontrados durante la importación**

##### **1. Error: "0 records imported"**

- Causa: Valores vacíos en columnas DECIMAL
- Solución: Uso de NULLIF() en el LOAD DATA

##### **2. Error: "Out of range"**

- Causa: Números demasiado grandes para DECIMAL(7,2)
- Solución: Cambio de tipo de dato a DECIMAL(20,2)

##### **3. Error: "Incorrect date value"**

- Causa: Fechas en formato "Jul 2013" o solo año "2019"
- Solución: Conversión a formato estándar YYYY-MM-DD con Python

##### **4. Error: "Incorrect integer value"**

- Causa: Columna id\_departamento vacía en CSV
- Solución: Generación aleatoria de IDs válidos con Python