

# INDEXACIÓN ESPACIAL

Juan David Barajas Camargo - 20191025074

27-05-2025

Bases de Datos Espaciales

# ¿Qué es la Indexación Espacial?

- Definición: "Estructura que organiza datos espaciales para acelerar búsquedas."
- Objetivo: Evitar escaneos secuenciales (lentos) en grandes conjuntos de datos.
- Ejemplo: Sin índice: 10,000 registros  $\times$  10,000 registros = 100,000,000 comparaciones. Con índice:  $\sim$ 20,000 comparaciones.



# ¿Qué es un índice espacial entonces?

01

Permite acelerar  
búsquedas  
espaciales  
(intersección,  
cercanía,  
contención, etc.)

02

En PostGIS se usa  
el índice GiST  
(Generalized  
Search Tree)

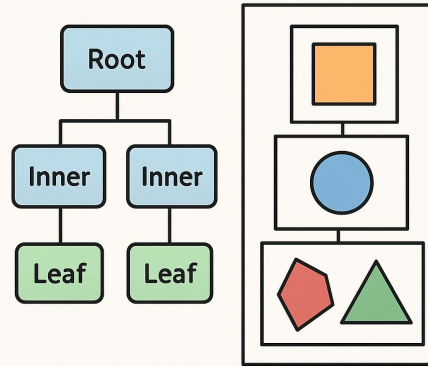
03

Similar a un índice  
tradicional, pero  
para geometrías

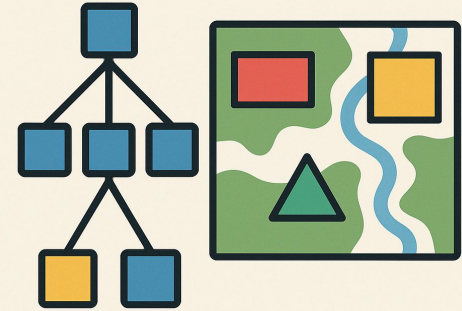
# ¿Cómo funciona el índice GiST?

- Crea una jerarquía de rectángulos (bounding boxes)
- Filtra con una primera pasada rápida (caja contra caja)
- Luego realiza la verificación exacta con funciones como ST\_Contains, ST\_Intersects, etc.

GiST Index – PostGIS



ÍNDICE GiST DE PostGIS

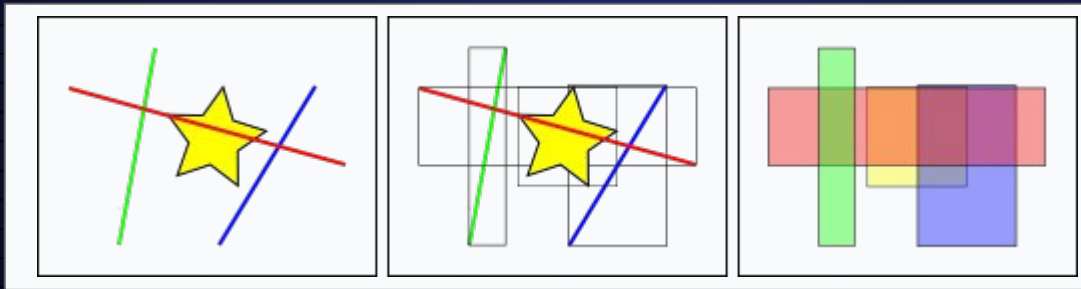


# Búsqueda por Bounding Boxes

- Los índices espaciales no indexan geometrías directamente, sino sus rectángulos envolventes (bounding boxes).

Proceso en 2 pasos:

1. Filtro rápido: ¿Qué bounding boxes intersectan? (usando índice).
2. Filtro exacto: ¿Qué geometrías intersectan realmente?





# R-Tree: La base de los índices espaciales

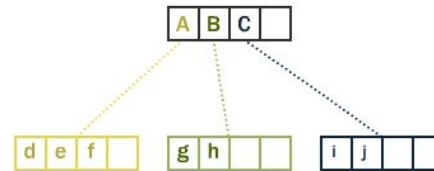
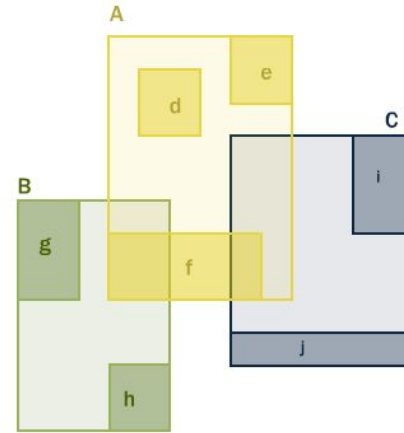
- Organiza datos en rectángulos anidados (jerárquicos).
- Autoajustable: Maneja densidad variable, superposición y tamaño de objetos.

Imagina que tienes un mapa de un país dividido en regiones, cada región en ciudades, y cada ciudad en barrios.

R-Tree funciona así:

- Nivel 1 (País): Todo el país es una "caja grande".
- Nivel 2 (Regiones): La caja grande se divide en cajas más pequeñas (ej. Norte, Sur, Este, Oeste).
- Nivel 3 (Ciudades): Cada región se divide en cajas de ciudades.
- Nivel 4 (Barrios): Cada ciudad se divide en cajas de barrios.

R-tree Hierarchy





# R-Tree: La base de los índices espaciales



## ¿Cómo se usa?

Si buscas "todos los parques en Bogotá", el R-Tree:

Descarta rápidamente las cajas de otras regiones (ej. Cali, Medellín).

Se enfoca en la caja de "Bogotá".

Dentro de Bogotá, busca solo los barrios que contienen parques.

## Resultado:

- Sin R-Tree: Buscarías manualmente en todo el país (lento).
- Con R-Tree: Enfocas la búsqueda solo donde importa (¡rápido!).

# Datos usados (caso real)



localidad	barrios	incidentes	Columns (44)
<div>Columns (8)</div> <ul style="list-style-type: none"><li>id</li><li>geom</li><li>LocNombre</li><li>LocAAdmini</li><li>LocArea</li><li>LocCodigo</li><li>SHAPE_Leng</li><li>SHAPE_Area</li></ul>	<div>Columns (18)</div> <ul style="list-style-type: none"><li>id</li><li>geom</li><li>objectid</li><li>codigo_loc</li><li>codigo_upz</li><li>codigo_bar</li><li>nombre</li><li>origen_ile</li><li>escala_cap</li><li>fecha_capt</li><li>codigo_b_1</li><li>formacion</li><li>actualizac</li><li>codigo_zon</li><li>responsabl</li><li>globalid</li><li>shape_area</li><li>shape_len</li></ul>	<div>Columns (23)</div> <ul style="list-style-type: none"><li>id</li><li>geom</li><li>FECHA</li><li>NUMERO_INC</li><li>ESTACION</li><li>SERVICIO</li><li>CLASE_DE_S</li><li>SUBCLASE_D</li><li>TIPO_SUBCL</li><li>USO_DE_LA_</li><li>CLASE_DE_U</li><li>SUBCLASE_1</li><li>CLASE_DE_V</li><li>SUBCLASE_2</li><li>TIPO_DE_VE</li><li>TIPO_DE_SE</li><li>MODALIDAD</li><li>TIPO_DE_MO</li><li>TIPO_DE_C</li><li>ÁREA_AFEC</li><li>ÁREA_EXPU</li><li>ORIGEN_DE_</li><li>CAUSAS</li></ul>	<div>Columns (44)</div> <ul style="list-style-type: none"><li>id</li><li>geom</li><li>OBJECTID</li><li>MVICCALZAD</li><li>MVICCAT</li><li>MVITIPO</li><li>MVINOMBRE</li><li>MVINALTERN</li><li>MVINPRINCI</li><li>MVINGENERA</li><li>MVINANTIGU</li><li>MVIETIQUET</li><li>MVISVIA</li><li>MVICIV</li><li>MVINVIA</li><li>MVIEVIA</li><li>MVIFRIGHT</li><li>MVITRIGHT</li><li>MVIFLEFT</li><li>MVITLEFT</li><li>MVITM</li><li>MVICM</li><li>MVINUMC</li><li>MVIACAL</li><li>MVIEFUN</li><li>MVITCONE</li><li>NAME</li><li>MUNICIPIO</li><li>PREFIJO</li><li>MVIVEL</li></ul>

Localidad: Rafael Uribe Uribe (Bogotá)

→ Tablas:

- ◆ localidad (polígono)
- ◆ barrios (polígonos)
- ◆ incidentes (puntos)
- ◆ vias (líneas)

<https://github.com/MzKaiser/Taller-14-BDE>



# ¿Por qué son cruciales los índices espaciales?

PostGIS los usa para optimizar operaciones espaciales (ej. ST\_Contains, ST\_Intersects).

**Caso práctico:** Consulta en la base de datos sin índice vs. con índice.



**Sin Índice**

300 ms



**Con Índice**

50 ms

# Importancia en PostGIS

## Manejo de Índices en PostgreSQL/PostGIS

**Eliminar índice:** `DROP INDEX IF EXISTS idx_vias_geom;`

**Crear un índice (Usando GIST):**

```
CREATE INDEX idx_vias_geom ON vias USING  
GIST(geom);
```

**Nota:** GIST (Generalized Search Tree) es un tipo de índice genérico en PostgreSQL que permite organizar datos complejos (como geometrías, textos, o rangos) para acelerar búsquedas.



# Creación del índice espacial

```
CREATE INDEX idx_vias_geom ON vias USING GIST(geom);  
CREATE INDEX idx_barrios_geom ON barrios USING GIST(geom);  
CREATE INDEX idx_incidentes_geom ON incidentes USING GIST(geom);
```



# Operador && para filtros por bounding box

- Útil cuando no se necesita precisión geométrica.

- **Ejemplo:**

```
SELECT b.nombre, COUNT(i.id)
```

```
FROM barrios b
```

```
JOIN incidentes i ON b.geom && i.geom -- Aquí usamos el operador &&
```

```
WHERE b.nombre = 'OLAYA'
```

```
GROUP BY b.nombre;
```

**Comparación:** && vs. ST\_Intersects (resultados diferentes por bounding box vs. geometría exacta).

# Optimización con ANALYZE y VACUUM

## Mantenimiento de Índices

“El archivista o la secretaria”

↓  
**ANALYZE**: Actualiza estadísticas para el optimizador de consultas.

Imagina que agregas 100 libros nuevos a una biblioteca. Si no actualizas el catálogo, el bibliotecario no sabrá dónde están.

ANALYZE es cómo actualizar el catálogo para que las búsquedas sean rápidas.

**Ejemplo:**  
ANALYZE barrios;

**Uso práctico:**

-- Después de cargar muchos datos, ejecuta:  
VACUUM ANALYZE barrios;

Esto sería como decirle a PostgreSQL: “Reorganiza y actualiza la info de los barrios de la localidad Rafael Uribe Uribe para que las consultas futuras sean más rápidas”.

**Autovacuum: Habilitado por defecto en PostgreSQL.**

“El limpiador o el conserje”

↓  
**VACUUM**: Recupera espacio liberado por UPDATES/DELETES.

Si borras páginas de un libro, quedan espacios en blanco. VACUUM reorganiza el libro para eliminar esos huecos y que no ocupe espacio innecesario.

**Ejemplo:**  
VACUUM ANALYZE barrios;



¡Gracias!