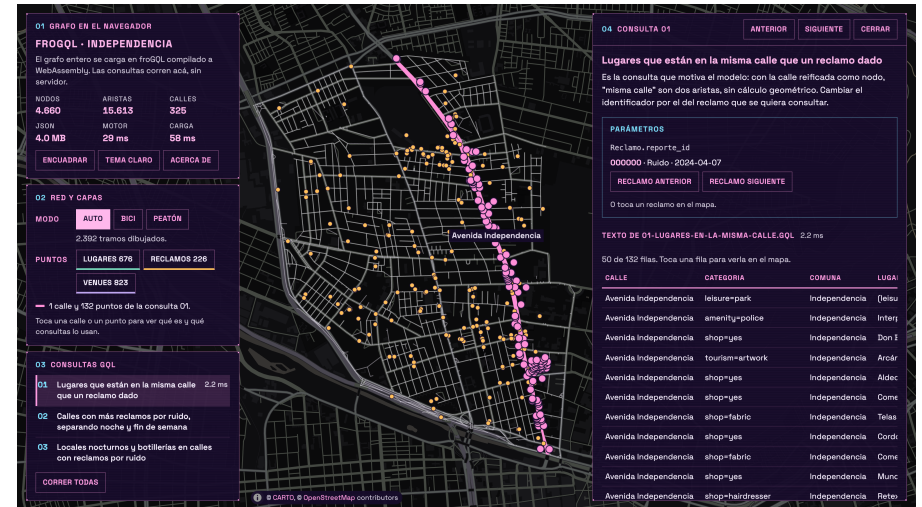


froGQL

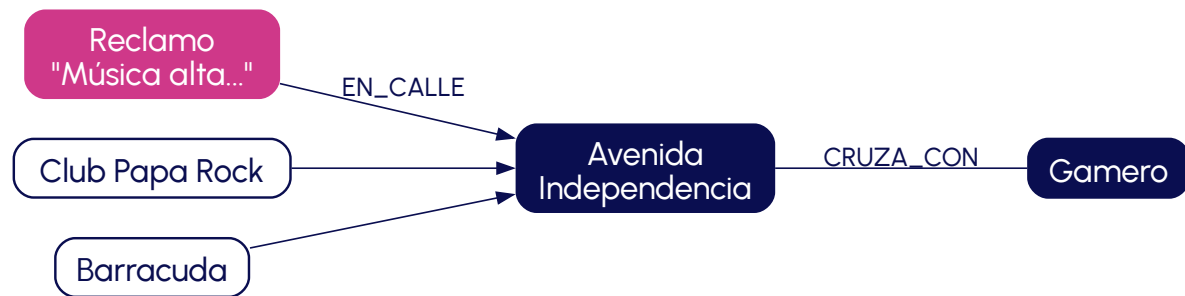
Una base de datos de grafos que corre
en el navegador

Departamento de Ciencias de la
Computación, Universidad de Chile

2026



froGQL guarda la comuna como un grafo que cabe en el teléfono



Un trozo del grafo de la comuna de Independencia.

- La base de Independencia y el motor pesan 960 KB y se descargan con la página web.
- Las consultas corren en el teléfono y responden en milisegundos, sin conexión.
- La desarrolla el DCC de la Universidad de Chile.

Un reclamo por ruido lleva a preguntar qué hay en su calle

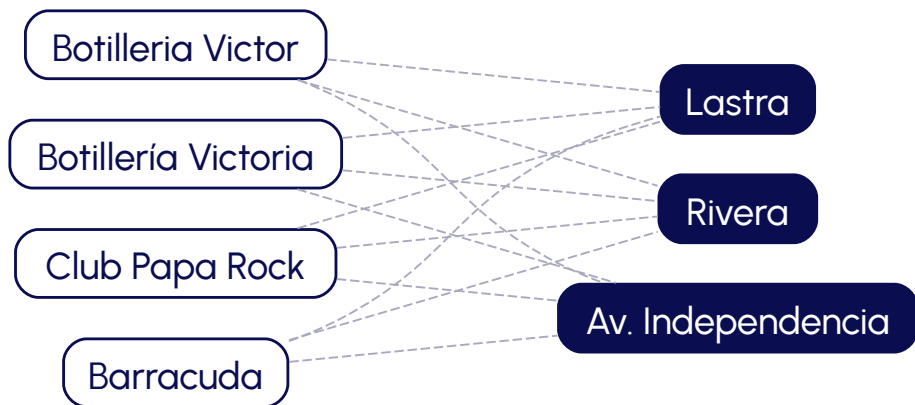
Música alta... 20.00... porfavor... no se puede estar tranquilo

Para responder hay que cruzar el reclamo de SOSAFE con los locales de OpenStreetMap.

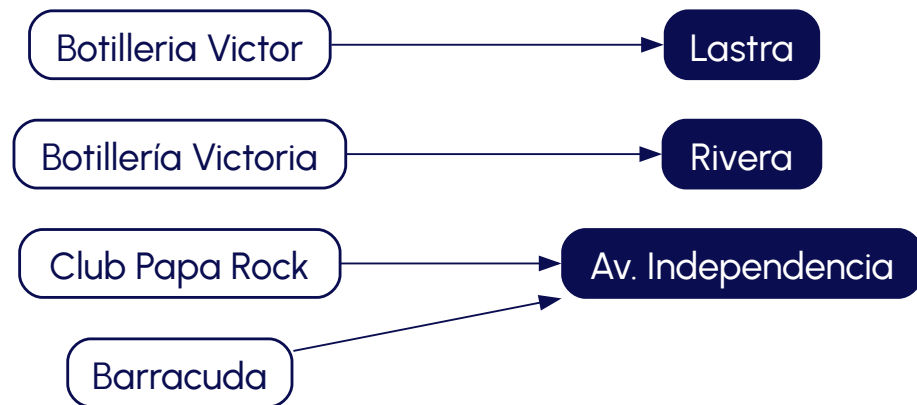


El grafo guarda la calle de cada local

Una planilla mide la distancia de cada local a cada calle.



El grafo guarda una arista `EN_CALLE` por local.



Con los 676 lugares y las 325 calles de Independencia, la planilla mide 219 700 distancias en cada pregunta. El grafo guarda 632 aristas, una por cada lugar con calle asignada.

En la calle del reclamo hay 132 lugares, entre ellos una discoteca y dos botillerías



El reclamo

está en

La calle

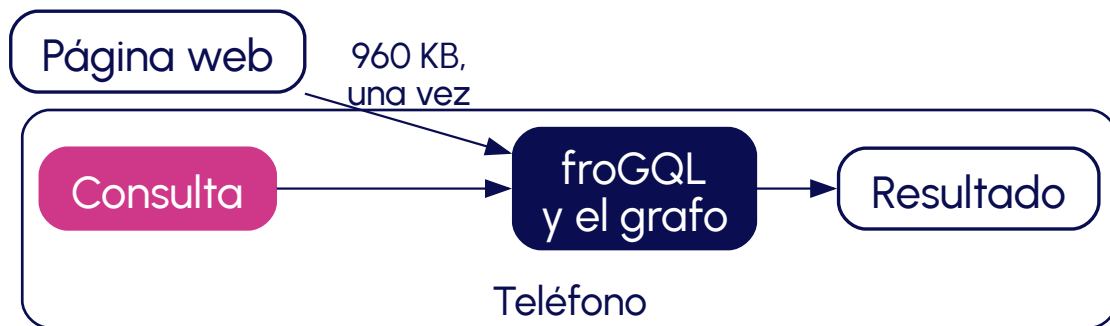
El local

está en

La base guarda a qué calle pertenece cada lugar, así que la consulta recorre dos aristas.

froGQL encontró los 132 lugares en 2 milisegundos.

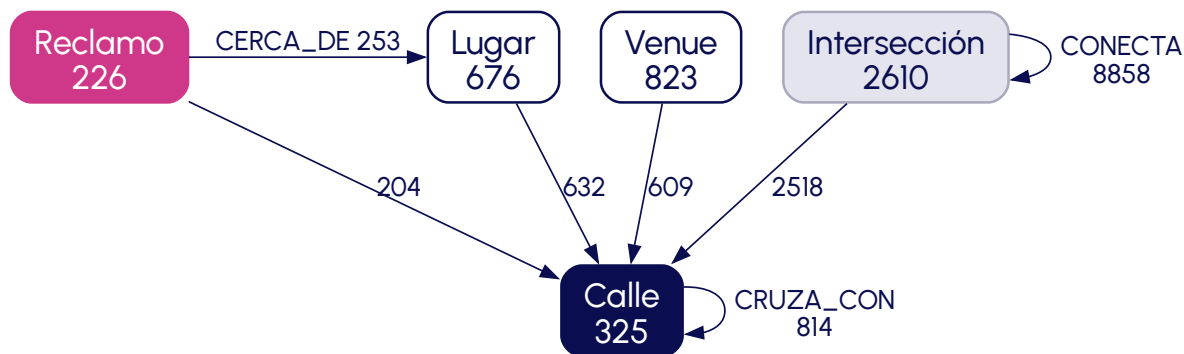
La página descarga la base y el motor una vez



Sin red, el mapa de fondo no carga, pero las consultas siguen respondiendo.



El grafo de Independencia tiene 4660 nodos y 13 888 aristas



Las flechas hacia `calle` son aristas `EN_CALLE`.
`CONECTA` suma las redes de auto, bici y peatón.

- Los datos son de OpenStreetMap, SOSAFE (abril de 2024) y Foursquare.
- La base y el motor pesan 960 KB comprimidos.
- Abrir la base toma 50 ms y la consulta del reclamo, 2 ms.
- En el Gran Santiago, con cien veces más nodos, esa consulta tarda lo mismo.

Cada constante de la consulta es un parámetro

- La categoría del reclamo, la calle y el tipo de local se eligen en pantalla.
- Un reclamo o una calle también se eligen tocándolos en el mapa.
- Cada cambio vuelve a correr la consulta.

En la versión de una columna, la consulta también se edita y se corre en vivo, sin rehacer la base.



The screenshot shows a dark-themed interface with a section titled 'PARÁMETROS' in red. It contains several interactive elements:

- Lugar.categoria en la lista:** Three buttons with checkboxes. The first two, 'amenity=bar' and 'amenity=pub', have red checkmarks. The third, 'amenity=nightclub', has an empty checkbox.
- shop=alcohol:** A button with a red checkmark.
- Agregar otro valor:** A text input field with a downward arrow on the right.
- Reclamo.categoria:** A dropdown menu currently showing 'Ruido (82)' with a downward arrow.
- Filas (LIMIT):** A dropdown menu currently showing '20' with a downward arrow.
- VOLVER A LOS VALORES DEL ARCHIVO:** A red button at the bottom.

01 GRAFO EN EL NAVEGADOR

FROGQL · INDEPENDENCIA

El grafo entero se carga en froGQL compilado a WebAssembly. Las consultas corren acá, sin servidor.

NODOS

4.660

ARISTAS

15.613

CALLES

325

JSON

4.0 MB

MOTOR

29 ms

CARGA

58 ms

ENCUADRAR

TEMA CLARO

ACERCA DE

02 RED Y CAPAS

MODO

AUTO

BICI

PEATÓN

2.392 tramos dibujados.

PUNTOS

LUGARES 676

RECLAMOS 226

VENUES 823

4 calles y 7 puntos de la consulta 03.

Toca una calle o un punto para ver qué es y qué consultas lo usan.

03 CONSULTAS GQL

01 Lugares que están en la misma calle que un reclamo dado2.2 ms

02 Calles con más reclamos por ruido, separando noche y fin de semana

03 Locales nocturnos y botillerías en calles con reclamos por ruido6.3 ms

CORRER TODAS

04 CONSULTA 03

ANTERIOR

SIGUIENTE

CERRAR

Locales nocturnos y botillerías en calles con reclamos por ruido

El filtro va por categoría, que combina el tag y su valor en una sola propiedad, porque los locales relevantes están repartidos en dos tags de OSM: `amenity` para bares y discotecas y `shop=alcohol` para las botillerías, que en Chile son el vector más directo de ruido en la vía pública. Filtrar solo por `amenity` las dejaba fuera. Restaurantes y comida rápida quedan fuera porque llenaban el resultado de cadenas que no explican el ruido. EXISTS resuelve la pertenencia sin unir cada local con cada reclamo de su calle, así que no paga el producto de las dos etiquetas.

PARÁMETROS

Lugar.categoría en la lista

☒ amenity=bar

☒ amenity=pub

☐ amenity=nightclub

☒ shop=alcohol

Agregar otro valor

Reclamo.categoría

Ruido (82)

Filas (LIMIT)

20

VOLVER A LOS VALORES DEL ARCHIVO

TEXTO DE 03-LOCALES-DE-ALCOHOL-EN-CALLES-CON-RUIDO.GQL6.3 ms

7 filas. Toca una fila para verla en el mapa.

CALLE

CATEGORIA

COMUNA

LUGAR

@ CARTO, @ OpenStreetMap contributors

La consulta de locales nocturnos y botillerías en calles con reclamos por ruido, sin las discotecas. Las calles y los locales del resultado quedan encendidos sobre la red de la comuna.

froGQL · zorzalerrante.github.io/frogql-case-study

9 / 13

Un patrón de largo variable mide cuánto hay que caminar

Con tanto ruido me dio sed... ¿dónde está la botillería más cercana?

```
MATCH ANY SHORTEST
  (r:Reclamo)-[:EN_ESQUINA]→(i)
  ~[e:CONECTA_PEATON]~+
  (j)←[:EN_ESQUINA]-(l:Lugar)
WHERE r.reporte_id = '000000'
  AND l.categoria IN
    ['amenity=bar', 'shop=alcohol']
  AND sum(e.largo_m) < 600
RETURN l.etiqueta, sum(e.largo_m) AS m
ORDER BY m
```

~+ camina las esquinas que haga falta y `sum(e.largo_m)` suma los metros.

Son 491 metros: ninguno de los 20 bares y botillerías de la comuna está más cerca.

En SQL, este recorrido requiere una consulta recursiva.



Caminando desde el reclamo, la botillería más cercana queda a 491 metros. Un salto entre calles con nombre no sirve para medir cercanía: la Avenida Independencia cruza con 58 calles y algunas miden tres kilómetros.

froGQL sirve cuando los datos caben en el dispositivo de quien consulta

QUÉ ES

Un motor de bases de datos de grafos desarrollado en el DCC de la Universidad de Chile. Usa GQL, el lenguaje que ISO estandarizó en 2024, y corre en Python, en JavaScript y en el navegador, sin servidor.

- Trabajo en terreno sin señal, como un catastro o una ronda de salud rural.
- Material educativo y guías de museo en lugares sin wifi.
- Datos sensibles que se consultan sin enviarlos a un servidor.

La base viaja dentro de la aplicación, que responde aunque falle la red.



El código QR abre la demo

Los scripts del repositorio descargan los datos, así que el caso se reproduce desde un clon.

```
pip install frogql · github.com/pleiad/frogql
```