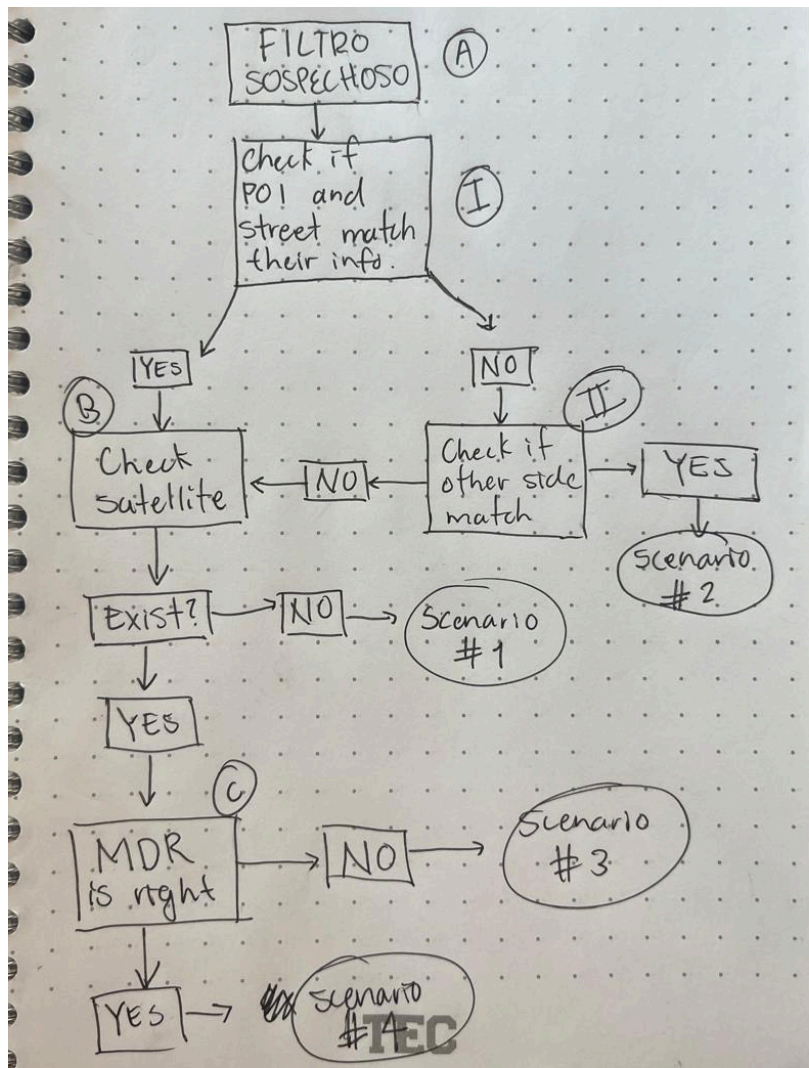


Empezamos leyendo la documentación y entendiendo los datos y el problema. Llegamos a la conclusión de que se tienen los puntos de interés (POI) los cuales pueden ser clasificados como sospechosos o no sospechosos, su clasificación depende de si se encuentran entre dos MDR como lo plantea el problema, una vez identificados estos puntos sospechosos pueden tomar una de las 4 resoluciones. El POI puede ser obsoleto y no existir, puede estar del otro lado de la calle, puede que la calle esté mal catalogada como MDR, o puede que esté POI sea una excepción a nuestras sospechas.

Analizando las reglas y los datos que se nos proporcionaron llegamos a el siguiente diagrama\*



Comenzamos limpiando los datos, eliminando columnas que no nos servían, dejamos los datos que consideramos relevantes para la actividad como las direcciones, código postal, coordenadas, el flujo, todo lo que nos diera indicios de ubicación, también mantuvimos las columnas que nos ayudan a

determinar si un camino es MDR o no. Esta primera etapa de limpieza de los datos se logró usando los códigos LIMPIEZA 1 2 y 3.

La primera fase del problema era detectar cuáles eran los POIs susceptibles a tener información errónea, siguiendo los lineamientos de la presentación dichos POIs deberían tener las siguientes características:

Estar entre dos MDR del mismo nombre a no más de 80 metros de distancia uno del otro

Para simplificar la búsqueda de la información se usó el código CONCATENAR para unir todos los tramos que compartían id y tener un control sobre los diferentes nombres que tenían estos tramos para referirse a el mismo camino.

Después se utilizó SENTIDOMULTI para añadirles el atributo que indica si dichos tramos son considerados MDR para luego eliminar todos los registros que tuvieran ese atributo como falso dejándonos así una base de datos que contenía todos los tramos correspondientes a una MADRE con sus correspondientes link\_id y todas las coordenadas correspondientes.  
Resultado: calles\_agrupadas\_consentido\_filtradasMulti

Por otro lado se usó RELACIONES para obtener la relación de todos los link\_id considerados MDR que también comparten nombres en común y por lo tanto eran candidatos para generar nuestros puntos posibles.  
Resultado: Relaciones.csv

Después con SOSPECHOSOS se realizó un análisis usando todos los POI relacionados hacía un link\_id dentro de los links antes mencionados, se tomaba el tramo al que estaba ligado y con las variables PERFECT se obtuvieron las coordenadas de los puntos ligados a estas calles, ahora se busco entre las calles con el mismo nombre (revisando el csv RELACIONES) para determinar si la calle más cercana estaba dentro del rango en el que el POI podría ser sospechoso según la documentación, ahora solo es cuestión de saber si la orientación está alineada para que el POI este en el centro de estas dos calles encontradas.

Debido a que el camino es un elemento unidimensional la orientación derecha o izquierda no cambia la distancia de nada por lo que podemos asignarle un movimiento perpendicular respecto al camino del POI de algunos 5 metros dependiendo en qué sentido está orientado el POI y respecto al cambio en las distancias podemos determinar si está dentro de la "zona sospechosa" o no, así obtenemos los POI que son sospechosos de tener errores. Fueron 49, una reducción bastante considerable ya que los POI totales son alrededor de 200,000 en la ciudad de México

## **Sección: Verificación de lado izquierdo o derecho de POIs en calles MULTIDIGIT**

Durante esta etapa del análisis, se aplicó un filtro adicional para validar si los POIs ubicados entre dos calles con la propiedad **MULTIDIGIT = "Y"** estaban efectivamente del lado correcto, de acuerdo con el valor declarado en el campo **POI\_ST\_SD** (Left o Right).

Para cada uno de los 49 POIs filtrados previamente (ubicados en zona de duplicidad de calles), se ejecutó el siguiente proceso:

1. Se identificó el **link\_id** de la calle a la que está asignado el POI.
2. Se obtuvieron todas las demás calles paralelas cercanas (a menos de 80 m de distancia) que compartieran el mismo nombre (**ST\_NAME**).
3. Se calculó en qué lado de la calle base (izquierdo o derecho) se encontraba la calle paralela más cercana, utilizando geometría vectorial y producto cruzado.
4. Se comparó esta orientación real con la que indicaba el POI en su campo **POI\_ST\_SD**.

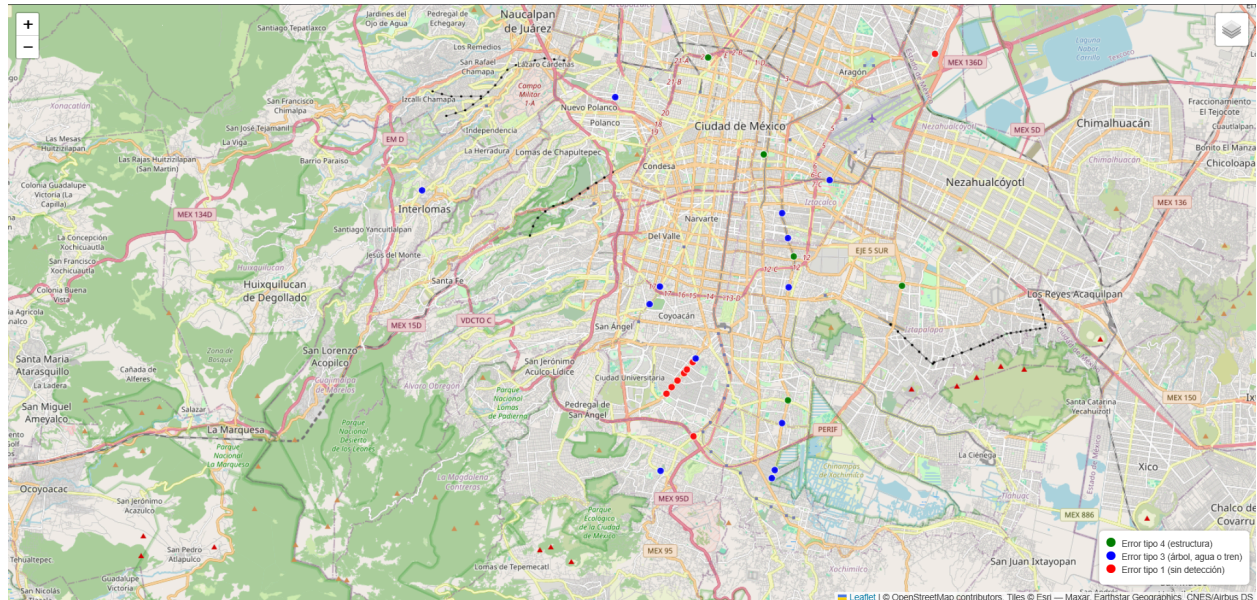
Como resultado de esta validación, se detectaron 8 POIs cuyo valor **POI\_ST\_SD no coincidía** con la orientación real determinada por la geometría de las calles. Es decir, el POI fue asignado como si estuviera en el lado izquierdo, pero se encontraba geométricamente del lado derecho (o viceversa).

## **Sección: Verificación de las detecciones de estructuras, agua y vías de tren**

Durante esta etapa se aplican diversas herramientas tecnológicas para identificar elementos relevantes en el mapa. Entre las más importantes se encuentran:

- El modelo YOLOv8n.pt: una red neuronal especializada en la detección de objetos en imágenes. Este modelo fue empleado para reconocer de manera automática la presencia de diferentes objetos, como estructuras, vegetación, cuerpos de agua y vías de ferrocarril.
- La API de imágenes satelitales de HERE: esta herramienta fue utilizada para poder generar imágenes con las latitudes y longitudes de cada punto de interés (POI) dentro del conjunto de datos. Esta API nos permitió obtener imágenes de una excelente calidad para que posteriormente fuera procesada por el modelo de detección.

Cada imagen analizada fue vinculada geográficamente con su punto de interés (POI) y visualizada por el visor interactivo. Además, cuando el modelo detecto alguno de los elementos mencionados, se genero una anotacion automatica del tipo de error o hallazgo, la cual tambien en el mapa mediante el codigo de color. Esto permitió verificar visualmente y de forma eficiente que POIs presentan características físicas relevantes en su entorno inmediato.



Clasificacion\_de\_sospechosos.csv X

HERE > HERE > here-visor > Clasificacion\_de\_sospechosos.csv > data

```
1  POI_ID,Observacion,Tipo_error
2  1134503129,estructura,Error tipo 4
3  1244535031,estructura,Error tipo 4
4  1244535031,estructura,Error tipo 4
5  1166530973,estructura,Error tipo 4
6  1210171349,ninguna,Error tipo 3
7  1244548091,ninguna,Error tipo 3
8  1227685676,ninguna,Error tipo 3
9  1227685692,ninguna,Error tipo 3
10 1220550251,ninguna,Error tipo 3
11 1161846481,ninguna,Error tipo 3
12 800030958,agua,Error tipo 3
13 1209392520,ninguna,Error tipo 3
14 1206599933,ninguna,Error tipo 1
15 1154482786,ninguna,Error tipo 3
16 1165481518,ninguna,Error tipo 3
17 1206970842,ninguna,Error tipo 3
18 1192816360,ninguna,Error tipo 1
19 1140348000,ninguna,Error tipo 3
20 1244241441,estructura,Error tipo 4
21 1217638479,ninguna,Error tipo 3
22 1225505886,ninguna,Error tipo 3
23 1217638482,ninguna,Error tipo 3
24 1225505897,ninguna,Error tipo 3
25 1217638478,ninguna,Error tipo 3
26 1217638483,ninguna,Error tipo 3
27 1217638480,ninguna,Error tipo 3
28 1217638481,ninguna,Error tipo 3
29 1139116420,ninguna,Error tipo 1
30 1023814513,ninguna,Error tipo 1
31 1175948951,estructura,Error tipo 4
32 1263005469,ninguna,Error tipo 1
33 1263005470,ninguna,Error tipo 1
```