

# Seguimiento 1



*Seguimiento tarea 1*

*Daniel Tomas Gallego*

## Objetivo:

Escribir un programa en Python que lea un archivo en formato GraphML (generado a partir de datos de OpenStreetMap) y construya un grafo en memoria utilizando la biblioteca SAX.

El grafo se presenta en un formato de lista de adyacencia y mantiene detalles sobre los nodos (id, osmid, lon, lat) y las aristas (src, dst, len), vitales para la futura Ejecución de algoritmos de búsqueda de rutas.

## Descripción del Código Fuente Abierto

### Clase GraphMLHandler (creargrafo.py)

GraphMLHandler, que hereda de xml.sax.ContentHandler, recibe eventos SAX para procesar el GraphML. Sus principales características son:

- **Inicialización:** Establece algunas estructuras para guardar el estado del proceso: key\_map (mapeo de clave a nombre de atributo), nodes (diccionario de nodos), edges (lista de aristas) y adjacency\_list (lista de adyacencia).
  - **Eventos SAX:**
    - startElement: Señala el inicio de elementos como key, node, edge y data, capturando sus atributos correspondientes (por ejemplo, id, source, target).
    - characters: Recoge el texto del elemento de datos.
    - endElement: Gestiona el cierre de elementos con la configuración de nodos o aristas, y también la actualización de las estructuras de datos.
  - **Procesamiento de Nodos:** Extrae osmid\_original, lon y lat; convierte lat y lon a punto flotante; captura lat y lon que son inválidos.
  - **Procesamiento de Aristas:** Guarda el source, target y length, y actualiza la lista de adyacencia para los nodos de destino y sus longitudes.
- Función parse\_graphml (creargrafo.py)  
Esta función configura el analizador SAX, establece el GraphMLHandler y analiza el archivo GraphML, devolviendo los diccionarios de nodos, aristas y la lista de adyacencia.
- Programa Principal (main.py)  
El script main.py, cuyo parse\_graphml lee el CR\_Capital.graphml y genera un archivo de texto (prueba.txt) que contiene los datos extraídos:
- Datos de Nodos con el ID correspondiente, OSMID, longitud y latitud.
  - Lista de Aristas con nodos y longitud.
  - Lista de Aristas, representando nodos conectados y las aristas entre ellos.

## Aspectos Destacados

- **Eficiencia:** Con SAX es posible procesar grandes archivos XML sin cargar el documento en memoria, lo que es adecuado para gráficos de alto volumen como los de OpenStreetMap.
- **Robustez:** La clase GraphMLHandler gestiona errores de conversión de datos (por ejemplo, coordenadas inválidas) lanzando excepciones para que la aplicación no se bloquee.
- **Representación del gráfico:** La lista de adyacencia generada permite que se ejecuten algoritmos de enrutamiento en trabajos futuros como la búsqueda del camino más corto.
- **Flexibilidad:** La representación es flexible en el sentido de que se pueden incluir más atributos para nodos o aristas si es necesario debido al mapeo dinámico de claves (key\_map).

## **Conclusión**

La Tarea 1 crea y representa con éxito en memoria un archivo de gráfico graphml. Estructura de Datos: El proceso es SAX y la eficiencia es segura.