



IIC2115 – Programación como Herramienta para la Ingeniería (II/2025)

Ejercicio Formativo 3 Capítulo 5

Aspectos generales

- **Objetivos:** aplicar los contenidos de análisis de redes.
- **Entrega:** lunes 20 de octubre a las 17:30 hrs. en el repositorio privado y respondiendo el ticket de salida.
- **Formato de entrega:** archivo Python Notebook (**E3.ipynb**) con el avance logrado para el ejercicio. El archivo debe estar ubicado en la carpeta **C5**. Utilice múltiples celdas de texto y código para facilitar el trabajo del cuerpo docente.
- **ULTRA IMPORTANTE:** todas las celdas utilizadas deben estar ejecutadas al momento de entregar el ejercicio, de modo que las salidas generadas sean visibles. En caso de no cumplir con esto, su entrega no será considerada como validación del ticket de salida.

Introducción

Con el fin de practicar los contenidos de análisis de redes, en este ejercicio deberá realizar una serie de procesamiento y visualizaciones de datos de la Región Metropolitana. No existe *a priori* un resultado correcto para cada misión.

Misiones

Para todas las misiones, considere la red vial obtenida a través de OpenStreetMap como fuente de datos y trabajada con la librería OSMnx. Ningún apartado va a considerar redes fuera de las **áreas urbanas del Gran Santiago**.

Nota: Para utilizar algoritmos como ruta mínima en redes que no provienen de osmnx se requieren agregar algunas componentes antes de crear el grafo a partir de un GeoDataFrame. Por ejemplo:

```
# Asignar ID unico a estaciones y agregar coordenadas x/y
estaciones_metro = estaciones_metro.reset_index(drop=True)
estaciones_metro['node_id'] = estaciones_metro.index
estaciones_metro['x'] = estaciones_metro.geometry.x
estaciones_metro['y'] = estaciones_metro.geometry.y
estaciones_metro = estaciones_metro.set_index('node_id')

# Definir nodos u y v para cada linestring (estaciones mas cercanas a inicio y fin)
def nearest_node(point):
    return estaciones_metro.geometry.distance(point).idxmin()
lineas_metro = lineas_metro.reset_index(drop=True)
lineas_metro['u'] = lineas_metro.geometry.apply(lambda g: nearest_node(Point(g.coords
    [0])))
lineas_metro['v'] = lineas_metro.geometry.apply(lambda g: nearest_node(Point(g.coords
    [-1])))

# Establecer indice multiple (u, v, key) y agregar longitud como peso
lineas_metro['key'] = 0 # Obligatorio para osmnx, 0 porque no se repiten arcos
lineas_metro.set_index(['u', 'v', 'key'], inplace=True)
lineas_metro['length'] = lineas_metro.geometry.length
```

1. Considere un viaje desde la **sala A1** del Campus San Joaquín de la Universidad Católica hasta la cima del **Cerro Chena** (-33.593891, -70.742777), combinando caminata (red peatonal) y metro. Encuentre la ruta más corta, indique su distancia y visualícela, diferenciando tramos por modo con colores distintos, además, muestre claramente las coordenadas de origen y destino. Finalmente estime el tiempo total de viaje de forma razonable en base a su criterio.

Hint: Para obtener la red peatonal puede usar parametros como `network_type='walk'`. Considere solo obtener la red para las comunas necesarias (San Bernardo y Macul)

2. Usando la red vial de la comuna de Ñuñoa:
 - Calcule la centralidad de intermediación (*betweenness*) y cercanía (*closeness*) para cada nodo.
 - Identifique los nodos en el percentil 90 en ambas métricas.
 - Visualice estos nodos junto con los puntos críticos de accidentes.

Realice un análisis visual sobre si existe coincidencia entre zonas críticas estructuralmente y zonas con alta inseguridad o accidentalidad.

¿Cambia el análisis al considerar la red del polígono que se obtiene al agregar un buffer de 2 km en torno a los bordes de la comuna? Argumente su respuesta.

3. Seleccione 4 distritos: con mayor densidad poblacional, mayor percepción de inseguridad, mayor número de robos con violencia o intimidación y mayor valor del suelo. Luego:

- Extraiga sus centroides.
- Genere una red completa entre ellos y calcule las distancias mínimas desde la red vial.
- Encuentre la ruta que los conecta todos con menor costo total y visualícela.

Hint: Considere crear una versión simplificada de la red solo con 4 nodos y los seis arcos que los conectan con sus respectivos largos (obtenidos a partir de la red original), luego aplicar un algoritmo simple sobre esta nueva red para resolver el problema y finalmente incorporar los resultados en la red original.

4. Utilizando la red vial de la comuna de Santiago, junto con las bases de datos georreferenciadas de accidentes y percepción de seguridad, conteste lo siguiente:

- Identifique y analice las zonas más críticas de la comuna desde una perspectiva de red. Para ello, seleccione e interprete al menos dos métricas de centralidad entre las vistas en clases (grado, intermediación, cercanía). Justifique su uso en este contexto.
- En base a estas métricas y a la distribución espacial de los eventos de accidentes e inseguridad, proponga al menos tres posibles ubicaciones para instalar un centro de control o intervención rápida (ej: carabineros, ambulancia, centro de monitoreo).
- Para cada ubicación propuesta, calcule su excéntrica en la red vial y discuta las implicancias de este valor en términos de cobertura territorial. ¿Cuál cree que ofrece mejor cobertura ante emergencias? ¿Cambiaría su elección inicial?

Hint: Puede utilizar funciones como `nx.degree centrality()`, `nx.betweenness centrality()`, `nx.closeness centrality()` y `nx.eccentricity()` para desarrollar los análisis correspondientes.

Nota: la excéntrica de un nodo corresponde a la mayor distancia (en número de pasos o longitud) entre ese nodo y cualquier otro en el grafo. Representa el peor caso de accesibilidad desde ese punto.

5. Considere que distintas zonas de la ciudad pueden presentar necesidades de conexión eficiente por razones como inseguridad, alta densidad poblacional o fomento del transporte activo.

- Identifique al menos 5 zonas prioritarias en el sector urbano de la RM basándose en criterios como: frecuencia de accidentes, baja percepción de seguridad o alto índice delictual. Justifique su elección apoyándose en los datos disponibles y visualizaciones.
- Genere una versión simplificada de la red vial que se componga de los caminos que conectan estas zonas (grafo completo ponderado por distancia) utilizando las rutas más cortas en la red original entre cada par de zonas seleccionadas.
- Genere un **Árbol de Expansión Mínima (MST)**, analice las diferencias con el grafo anterior y discuta las implicancias de usar una red mínima frente a una red con caminos redundantes. ¿Qué ventajas y riesgos presenta cada uno en el contexto urbano?

Hint: Puede utilizar `nx.shortest_path_length()` para obtener las distancias y `nx.minimum_spanning_tree()` para generar el árbol.

Nota: un Árbol de Expansión Mínima (MST) es una subred que conecta todos los nodos del grafo sin formar ciclos y con el menor peso total posible. Representa una forma eficiente pero no redundante de conexión.

6. Considere una red simplificada donde cada nodo representa una de las siguientes comunas del Gran Santiago: Lo Barnechea, Vitacura, Las Condes, Providencia, Ñuñoa, La Reina, Peñalolén, Macul, La Florida y Puente Alto. Modele la red vial que conecta estas comunas utilizando solo calles principales y autopistas extraídas desde OpenStreetMap.

- Modele el grafo intercomunal conservando únicamente los segmentos viales principales (calles primarias). Estime la **capacidad de cada arista** y normalice los valores para usarlos como atributo de capacidad. Justifique brevemente su metodología.
- Seleccione como **comuna origen** aquella con mayor nivel de delitos y como **comuna destino** aquella con mejor percepción de seguridad, según los datos disponibles. Calcule el **flujo máximo** entre ambas e interprete el resultado: ¿la red tiene suficiente capacidad para canalizar una posible demanda en ese trayecto?
- Identifique los **cuellos de botella** en la red y analice qué aristas limitan el flujo. Discuta cómo podría rediseñarse la red para mejorar su resiliencia ante bloqueos o alta demanda.

Hint: Puede utilizar `nx.edge_betweenness centrality()` para estimar el uso relativo de cada conexión y luego aplicar `nx.maximum_flow()` y `nx.minimum_cut()` para el análisis de capacidad y vulnerabilidad. Se recomienda visualizar la red resultante y destacar los segmentos críticos.

Nota: el flujo máximo representa la mayor cantidad posible de "flujo" (como vehículos, personas o información) que puede pasar de un nodo origen a uno destino a través de la red, respetando las capacidades. El corte mínimo identifica el conjunto de conexiones críticas cuya remoción limitaría completamente ese flujo.