Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2115 - Programación como herramienta para la ingeniería

Análisis de redes

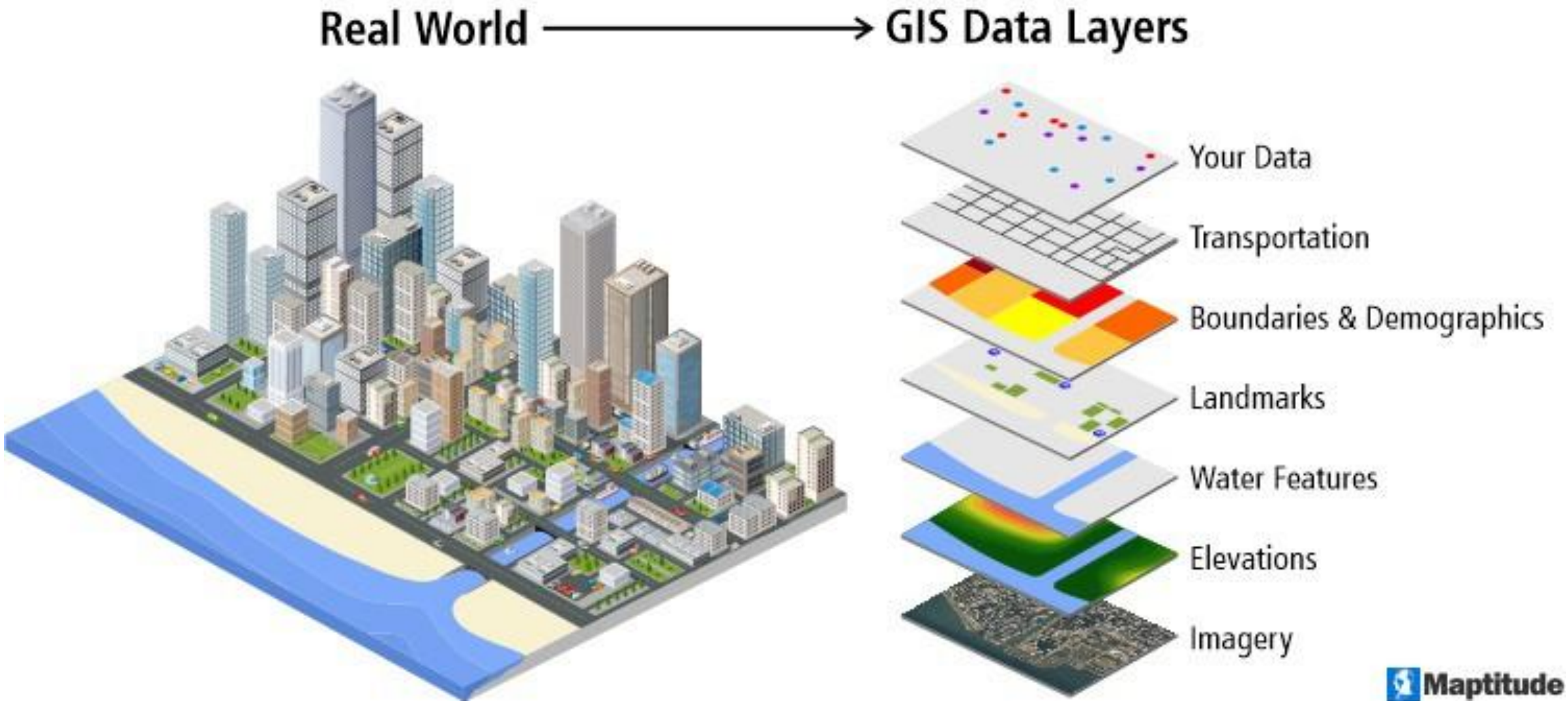
Profesor: Hans Löbel

¿Cómo podríamos encontrar la mejor ruta de evacuación en caso de inundación?

(manteniendo la idea del análisis de datos geoespaciales)

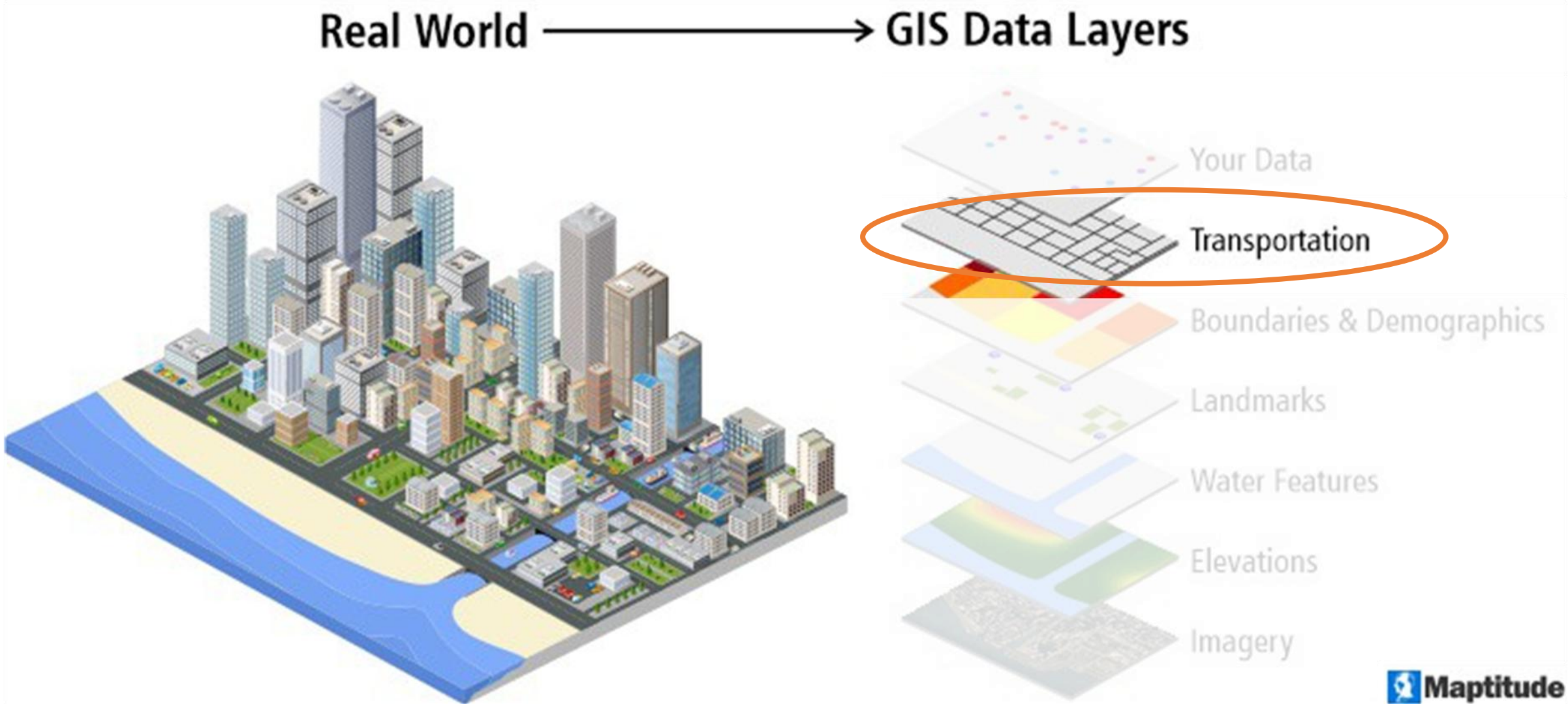
¿Cómo podríamos encontrar la mejor ruta de evacuación en caso de inundación?

(manteniendo la idea del análisis de datos geoespaciales)



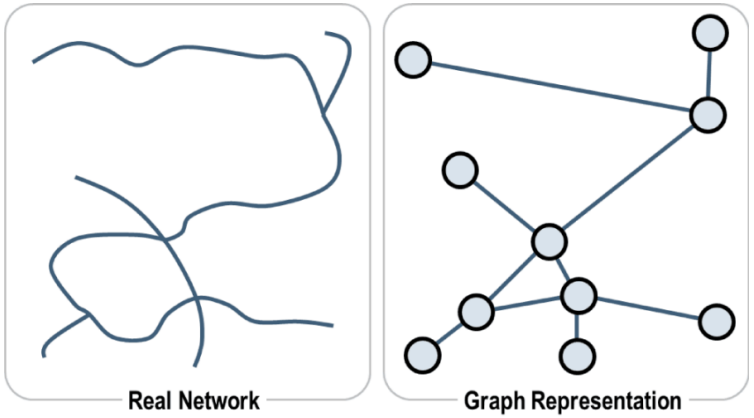
¿Cómo podríamos encontrar la mejor ruta de evacuación en caso de inundación?

(manteniendo la idea del análisis de datos geoespaciales)



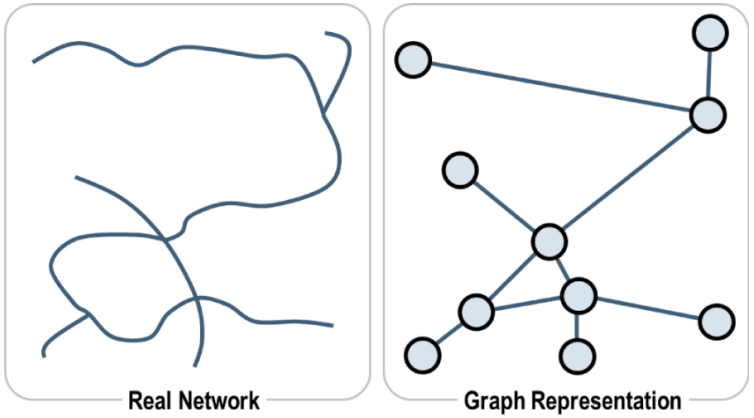
No nos basta con la información geométrica de la red de transporte/vial

- Necesitamos una abstracción más fácil de operar que los GeoDataframes.
- Grafos son una solución eficiente para esto.
- Un grafo es una abstracción que nos permite representar redes, como la de transporte.
- Por ejemplo, cada punto de interés o intersección se modela como un nodo, y cada calle que los conecta como una arista.



No nos basta con la información geométrica de la red de transporte/vial

- Con un grafo podemos analizar preguntas como las siguientes:
  - ¿Cuál es la ruta más corta entres dos puntos de la ciudad?
  - ¿Qué puntos de la ciudad son más importante para la red de transporte?
  - ¿Dónde se satura el flujo vehicular si hay un corte?
- Para responderlas, existen múltiples algoritmos para grafos: ruta mínima, flujo, ruteo, centralidad, entre otros.
- Nos basaremos principalmente en 2 librerías: NetworkX y OpenStreetMap (además de GeoPandas).

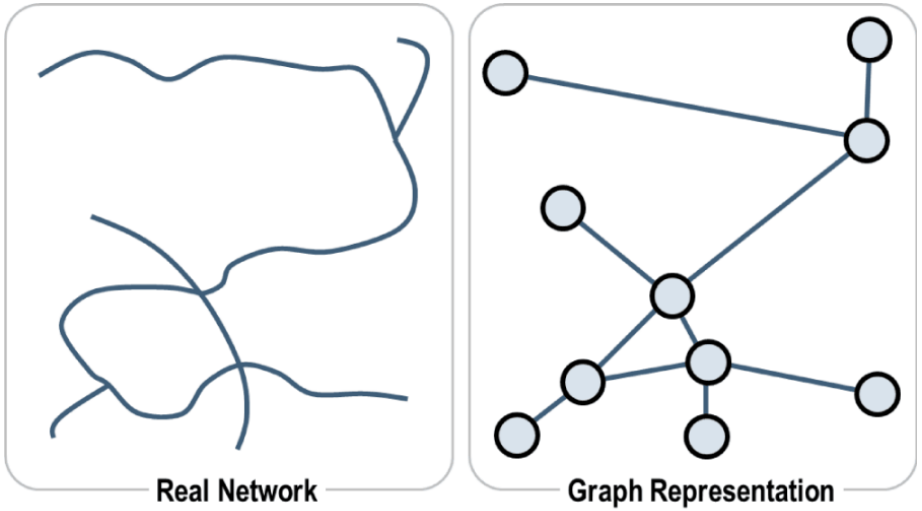


NetworkX

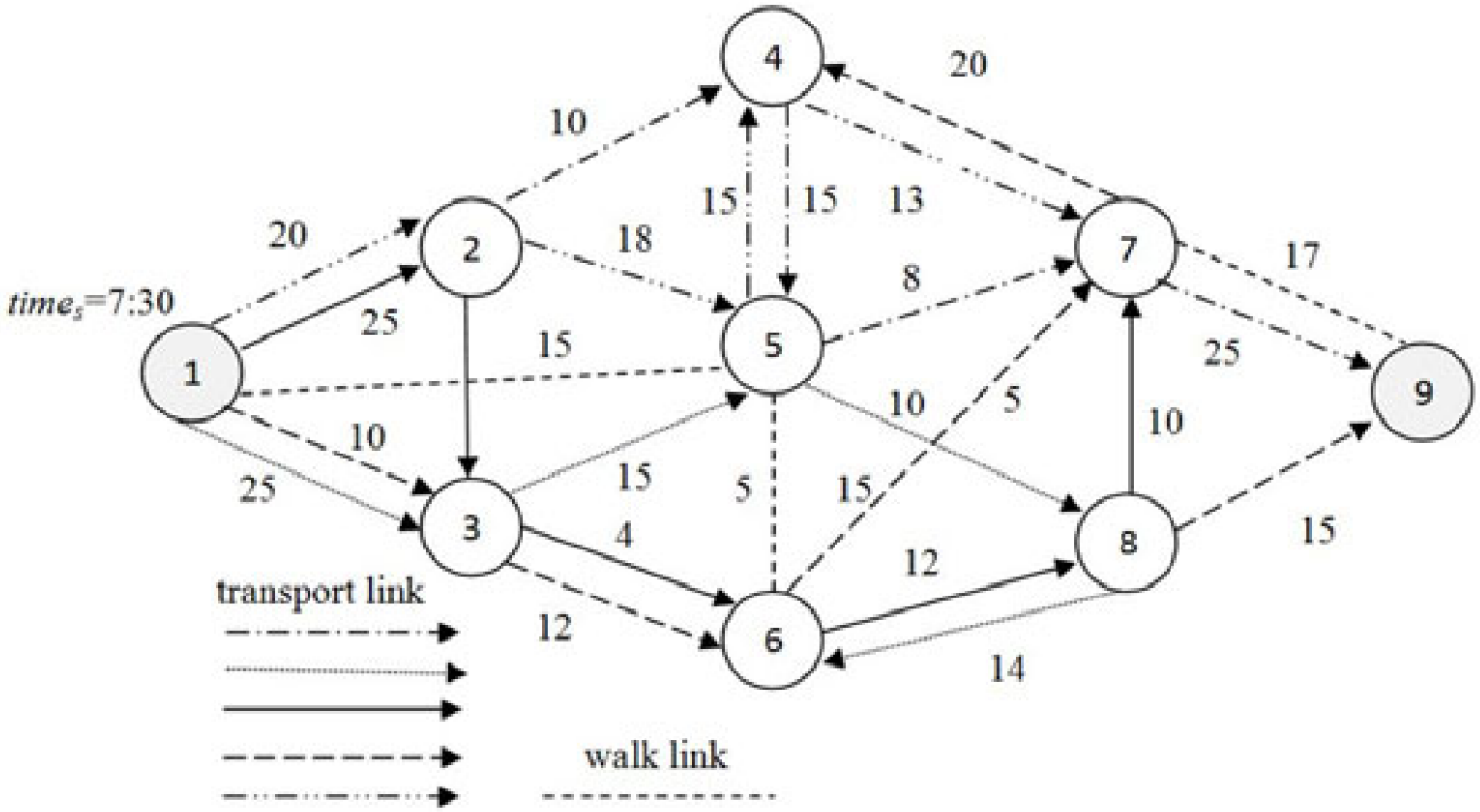
OpenStreetMap

Más formalmente, ¿qué es un grafo?

- Abstracción matemática para representar redes.
- Consta de nodos y arcos, que representan entidades y las relaciones entre ellas, respectivamente.
- Aplicable a múltiples dominios: redes de transporte, redes sociales, redes de comunicación, moléculas, etc.
- Para redes de transporte, podemos utilizar distintos grafos dependiendo del problema.

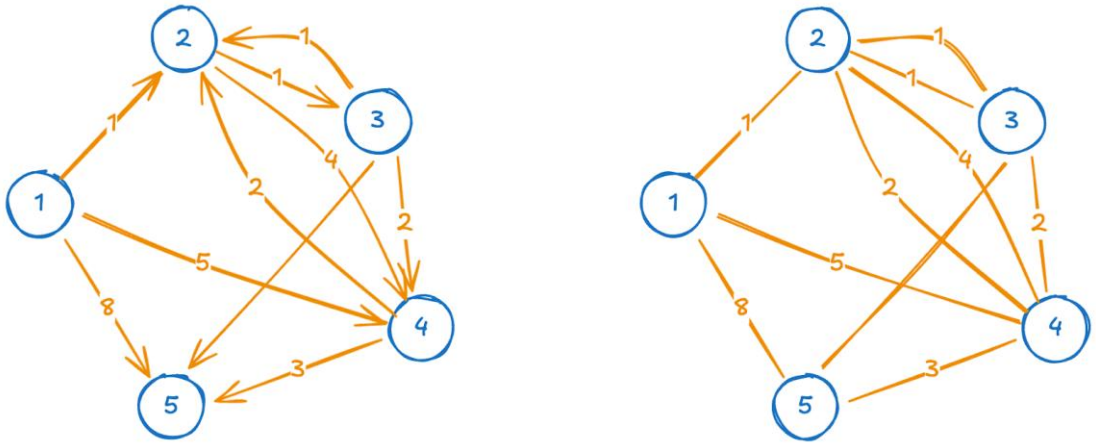


La gran mayoría de las veces nos interesará obtener/generar el grafo adecuado al problema y luego analizarlo o movernos por él

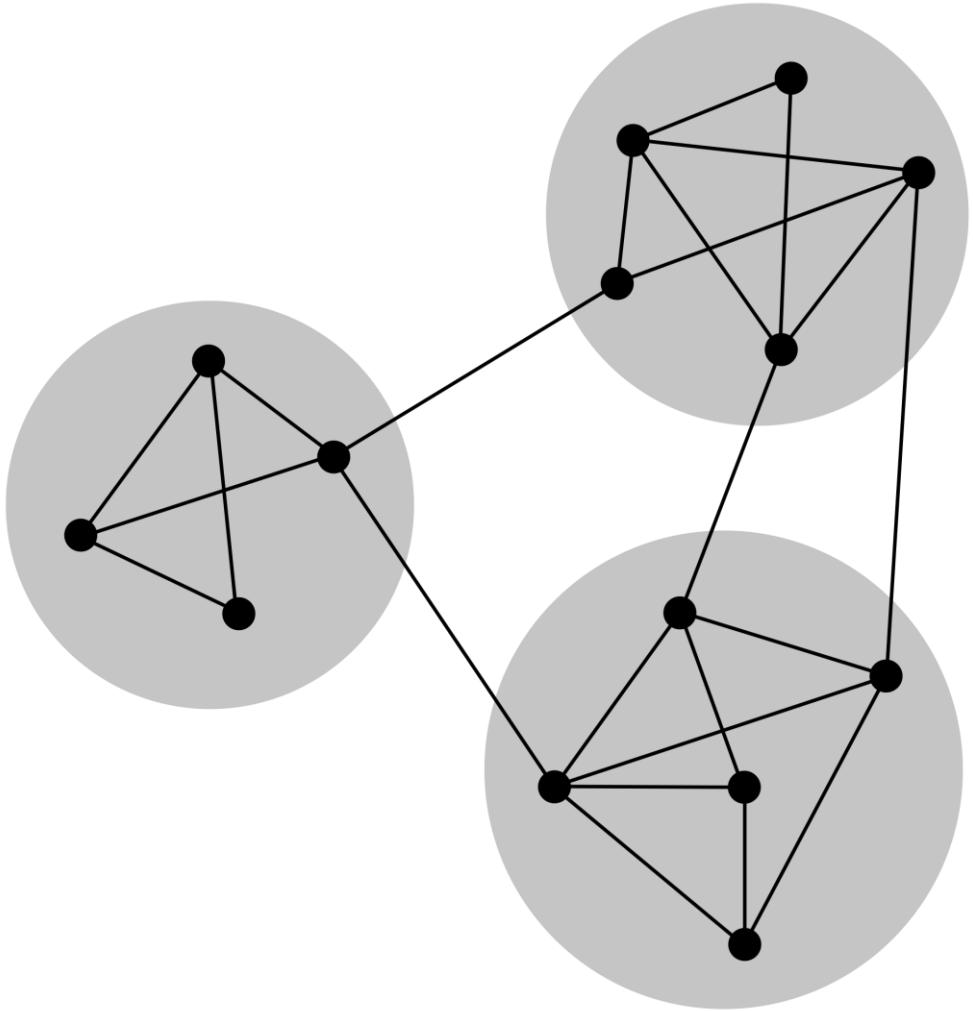


Por ejemplo, en el mundo real, algunas conexiones son mutuas y otras tienen sentido único

- Si pensamos en una red vial como peatones, típicamente podemos movernos en cualquiera de los dos sentidos en una vereda.
- Sin embargo, si lo pensamos como conductores de un vehículo, esto no es adecuado: calles pueden tener tienen un solo sentido.
- Para modelar esto, utilizaremos dos tipos de grafos distintos:
  - Dirigido: las aristas tienen una dirección específica, representando relaciones unidireccionales.
  - No dirigido: las aristas no tienen dirección, representando relaciones bidireccionales.

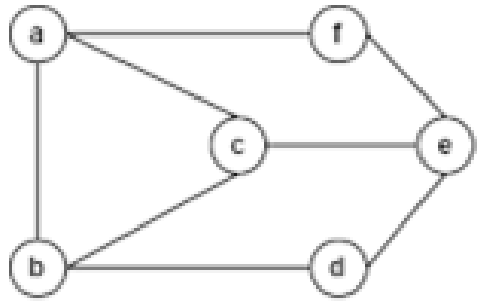


Otra pregunta interesante tiene que ver con el estudio de la “resiliencia” de una red

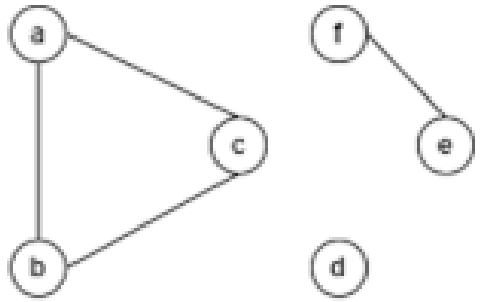


Otra pregunta interesante tiene que ver con el estudio de los puntos críticos de un grafo

- La conectividad de un grafo mide cuán interconectados están los nodos en este.
- Un grafo conexo es aquel donde existe un camino desde cualquier para de todos. Un grafo que no cumple con esto es disconexo.
- La conectividad de un grafo G, $k(G)$, se refiere a la cantidad mínima de nodos o arcos que deben eliminarse de G para hacerlo disconexo.
- Así, mientras mayor sea la conectividad de un grafo, mayor es su resiliencia.



Conexo

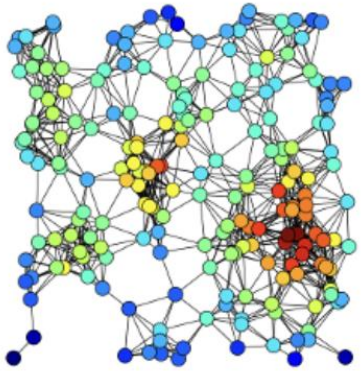


No conexo con 3 componentes conexos

Las medidas de centralidad nos permiten hacer un análisis de la importancia de los nodos

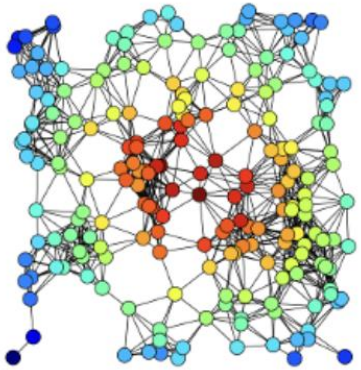
- Centralidad de grado: mide cuántas conexiones directas tiene un nodo. Nodos con alta centralidad de grado tienen mayor influencia inmediata en la red.

$$C_D(v) = \deg(v)$$



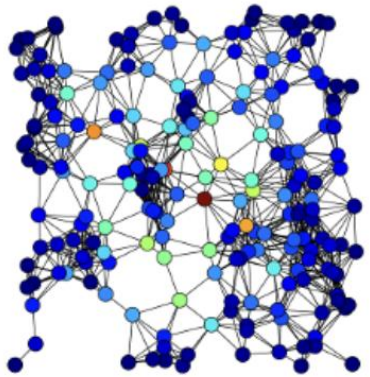
- Centralidad de cercanía: mide la distancia promedio de un nodo a todos los demás. Nodos con alta centralidad de cercanía pueden llegar rápidamente a otros nodos.

$$C_C(v) = \frac{n-1}{\sum_{u \neq v} d(v,u)}$$



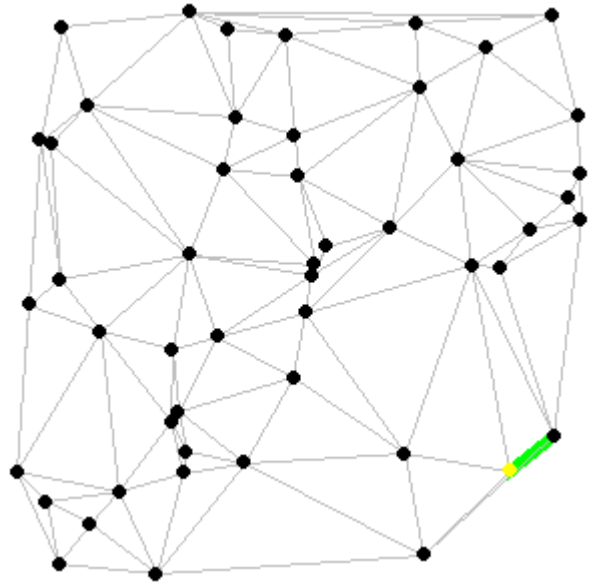
- Centralidad de intermediación: mide cuántos caminos más cortos pasan a través de un nodo. Nodos con alta intermediación controlan el flujo de información.

$$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$



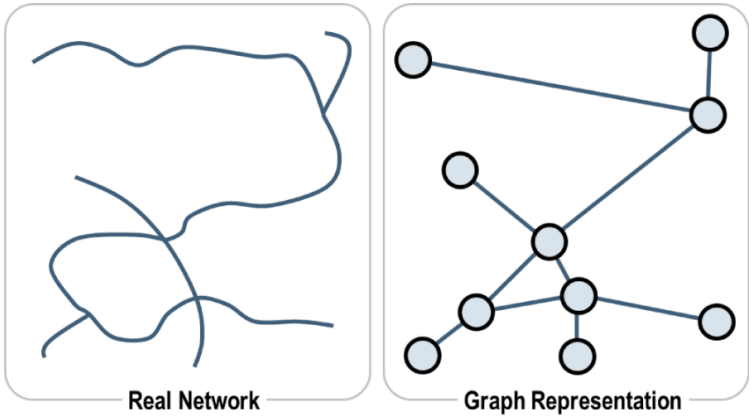
Desde un punto de vista práctico, para analizar los grafos, utilizaremos algoritmos conocidos y eficientes, ya implementados en la librería NetworkX

- Ruta óptima
- Vendedor viajero
- Clique
- Cortes mínimos
- Flujo
- Y muchos más…



¿Qué tipo de ejercicios considera este capítulo?

- El flujo comienza típicamente con la identificación del lugar a analizar y el tipo de red a descargar de OpenStreetMap.
- Posteriormente, se localizan los puntos de interés, típicamente como nodos en la red.
- Luego, aplicamos alguno de los algoritmos disponibles, dependiendo de lo que se busque.
- Finalmente, se grafica el resultado de las operaciones.



NetworkX

OpenStreetMap

Identificación de lugar de análisis y descarga de la red

Podemos obtener el grafo/red de interés, de múltiples maneras:

- Nombre del lugar
- Polígono
- Bounding Box
- Centro y radio

Adicionalmente, es posible especificar qué tipo de vialidad tendrá la red.

```
G = ox.graph_from_place('Macul')
ox.plot_graph(G, figsize= (20,20), bgcolor = 'w', node_color = 'red', edge_color = 'black');
```

Con poco esfuerzo, luego es posible ubicar los nodos y ejecutar algún algoritmo

```
orig_node = ox.distance.nearest_nodes(G, -70.746707, -34.164603)
dest_node = ox.distance.nearest_nodes(G, -70.711568, -34.180152)
route = nx.shortest_path(G, orig_node, dest_node, weight = 'length')
ox.plot_graph_route(G, route, figsize=(20,20), bgcolor = 'w', node_color = 'black', edge_color = 'black');
```

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2115 - Programación como herramienta para la ingeniería

Análisis de redes

Profesor: Hans Löbel