

# TALLER 2

**7 OCTUBRE 2024**

**MARTINA.BAQUEDANO@UC.CL**

# GRAFO A TRABAJAR

**north = -41.4**

**south = -41.496**

**east = -72.85**

**west = -73.03**

**G = ox.graph\_from\_bbox(north,south,east,west,network\_type='drive')**

**Recomiendo fuertemente trabajar con este grafo, ya que garantiza que todos los nodos esten incluidos en el grafo**

**Su grafo se deberia ver de la siguiente forma:**

# GRAFO A TRABAJAR



# TIPS PARA MODELAR

Los grafos de OSMNX traen muchos arcos paralelos. Esta tarea es de rutas mínimas, por lo cual no hay problema en que eliminen arcos paralelos; se espera que se queden con los de menor distancia.

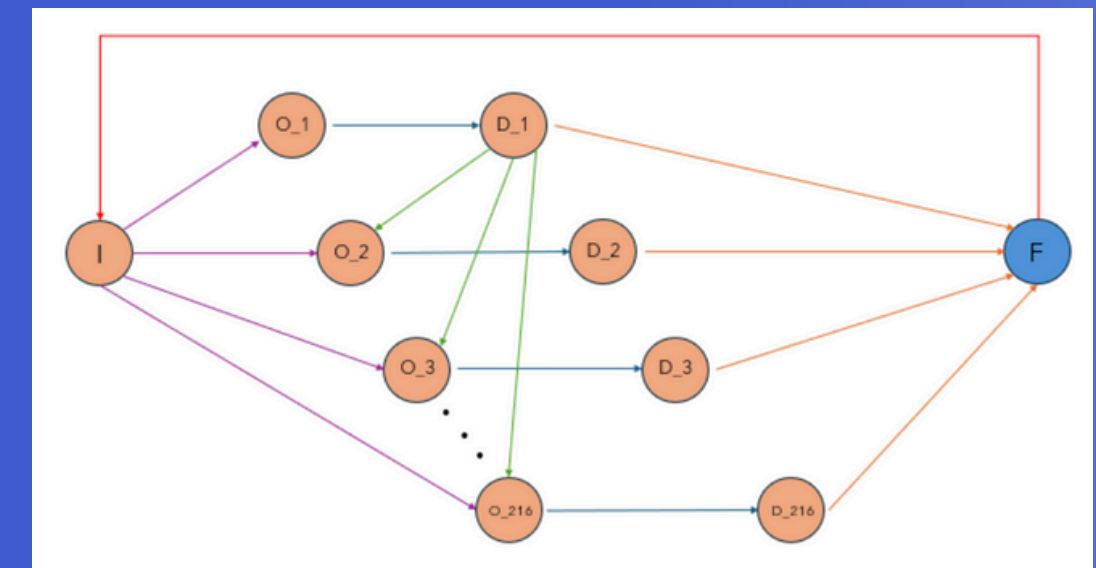
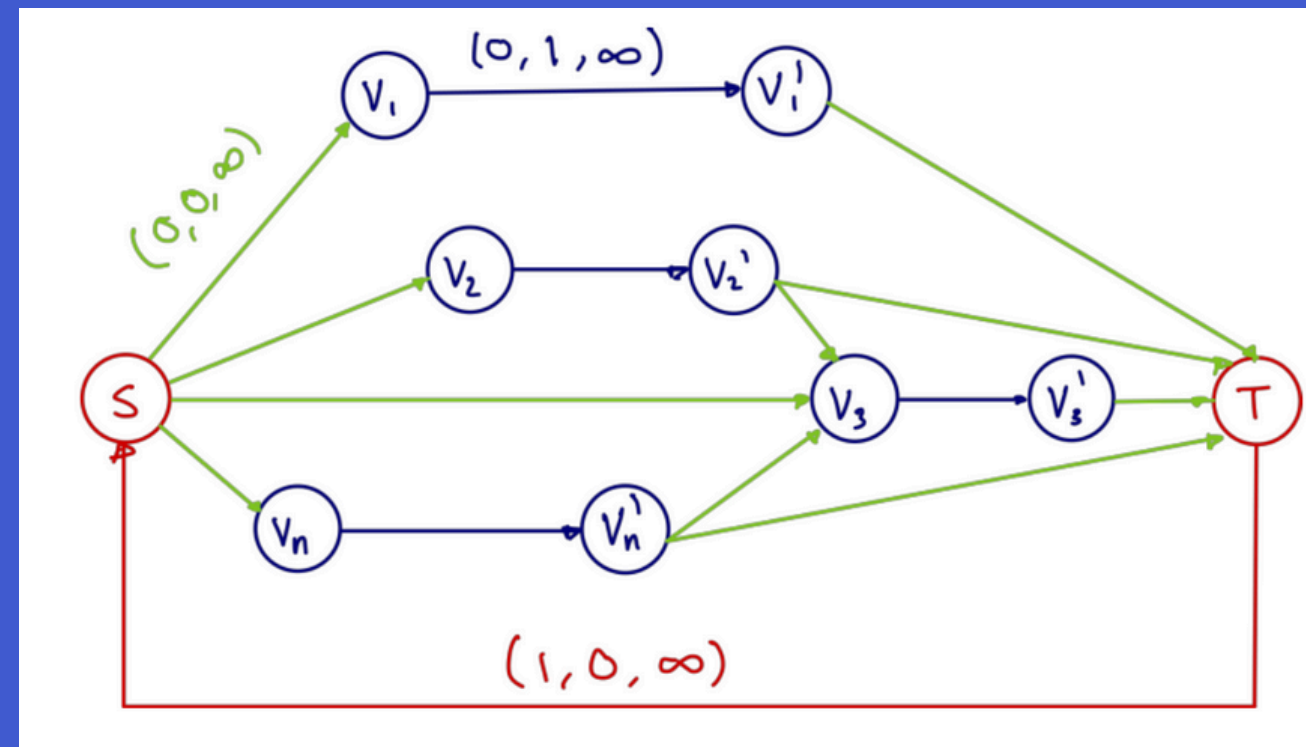
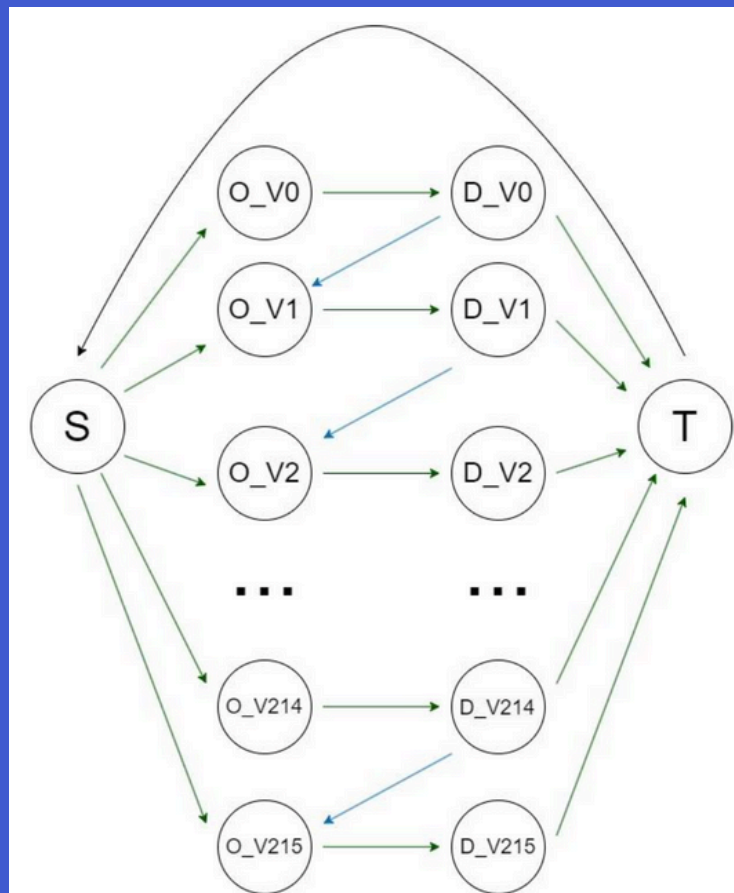
```

duplicated = []
for u,v,a in G.edges(data=True):
    if len(G[u][v]) == 2:
        duplicated.append((u,v,1))
    elif len(G[u][v]) == 3:
        duplicated.append((u,v,1))
        duplicated.append((u,v,2))
duplicated = list(dict.fromkeys(duplicated))
G.remove_edges_from(duplicated)
```

# TIPS PARA MODELAR

Es recomendable que guarden su grafo original, sin modificaciones, para graficar.

Prueben su PPMC en una instancia pequeña, esta es la misma que recomiendo poner en su informe. Ejemplos (distintos a los de esta tarea):





# MANEJO DE LA INSTANCIA

Codigo con el que pueden descargar datos directo en su colab (puede colapsar y dejar de funcionar, si pasa me avisan y les hago otro):

```
!wget https://www.dropbox.com/scl/fi/ulpuv76wq3t6706rdeiac/Instancia_T2_2024-2.zip?rlkey=ainq96fdxh0xifo82n3le2n55&st=m3hamp96&dl=0
```

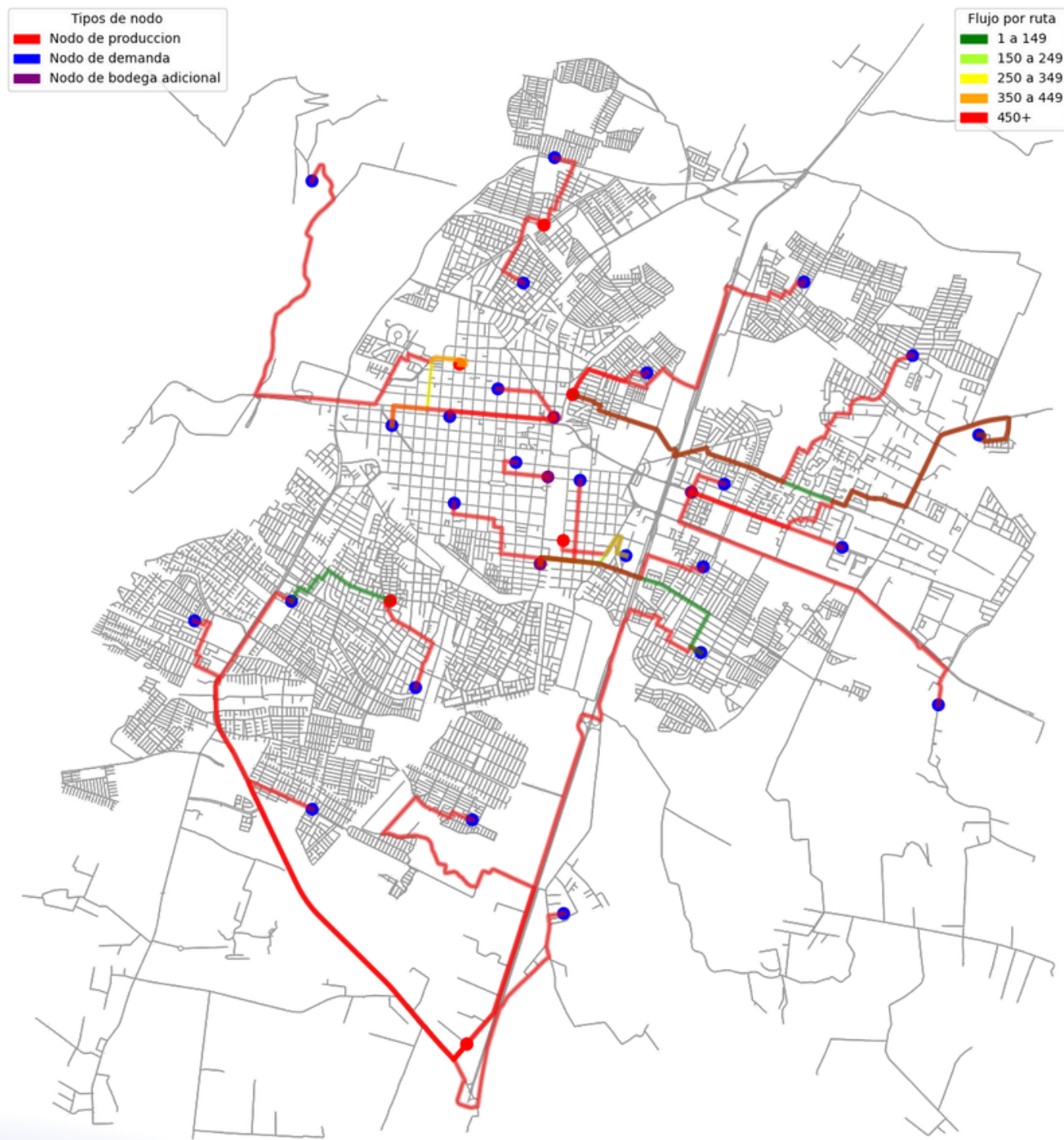
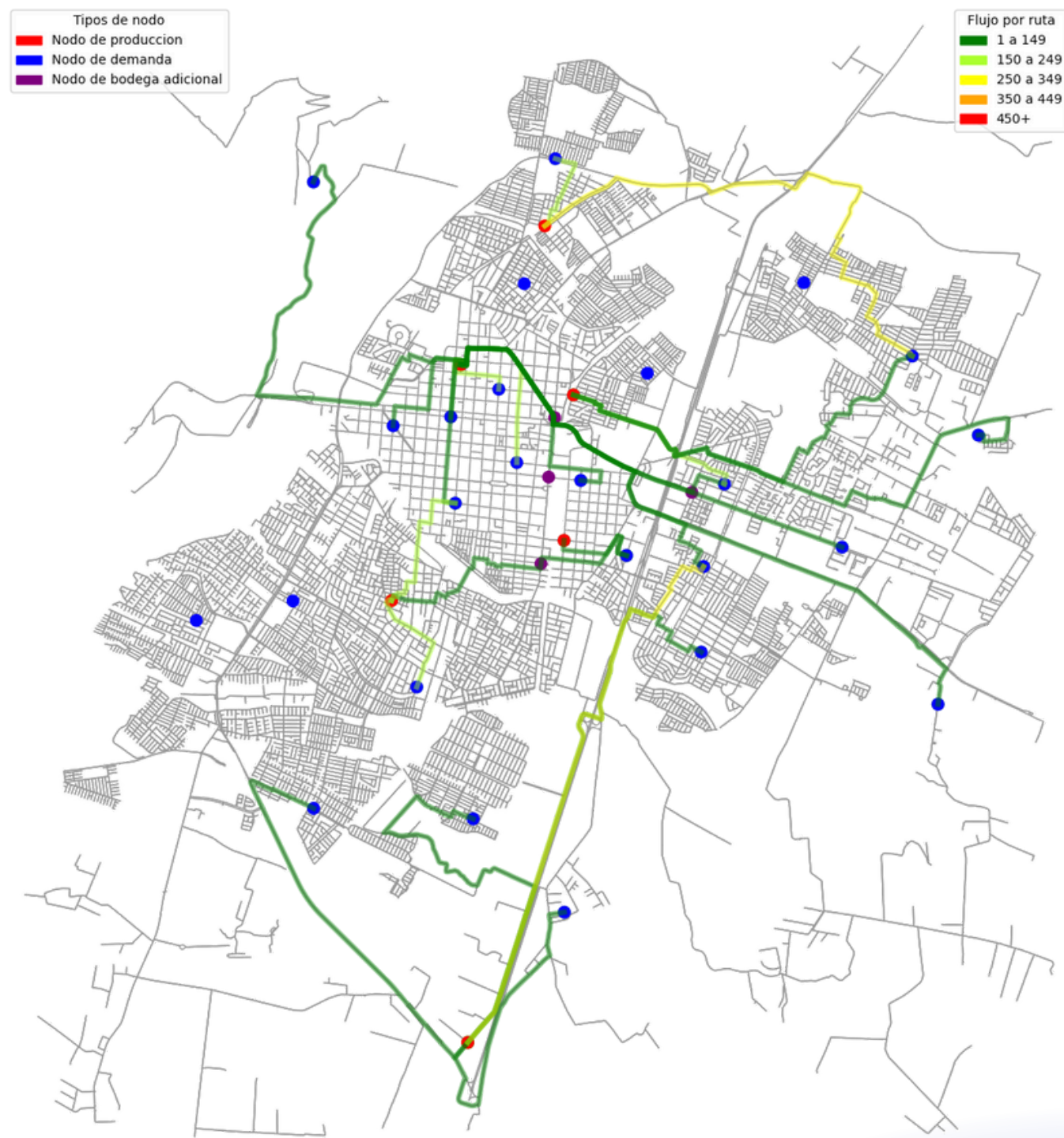
```
!unzip Instancia_T2_2024-2.zip?rlkey=ainq96fdxh0xifo82n3le2n55&st=m3hamp96&dl=0
```

```
!rm Instancia_T2_2024-2.zip?rlkey=ainq96fdxh0xifo82n3le2n55&st=m3hamp96&dl=0
```

Como recomiendo leer los .cvs:

```
demanda_max = pd.read_csv('/content/Instancia_T2_2024-2/demanda_max.csv',  
index_col=0)
```

# GRÁFICO QUE MUESTRE LAS PLANTAS, BODEGAS Y MINORISTAS, Y QUE ILUSTRE LAS RUTAS Y CARGAS TRANSPORTADAS ENTRE ESTOS NODOS



# NODOS Y ARCOS DE COLORES

Para darle un color en particular a un nodo u arco, cuando es graficado, los pasos que recomiendo son:

1. Hacer una lista que guarde los colores. Ej: `colores_nodos = []`
2. Iterar sobre todos los nodos o arcos
3. Por cada nodo añadir como elemento a la lista el color que debe tener. Ej: Verifique un nodo y se que es de produccion, entonces `colores_nodos.append("rojo")`

No color, o color transparente es: `"#FF000000"`



# LEYENDAS

Importar:

```
import matplotlib.pyplot as plt
```

```
import matplotlib.patches as mpatches
```

Para crear una leyenda en un grafo se usa:

```
patches = []
```

```
patches.append(mpatches.Patch(color='grey', label='Nodos A'))
```

```
patches.append(mpatches.Patch(color='blue', label='Arcos B'))
```

```
patches.append(mpatches.Patch(color='red', label='Nodos C'))
```

y luego antes de `plt.show()` poner:

```
plt.legend(handles=patches)
```

# **EJEMPLO PLAN OPERATIVO DIARIOS**

**Día X:**

- **Enviar 35 dulces de la bodega 2 al almacén 3.**
- **Producir 75 dulces en la planta 7.**
- **Enviar 56 dulces de la planta 4 a la bodega adicional 3.**
- **...**

# ENCONTRAR CAMINOS MAS CORTOS EN PUERTO MONTT

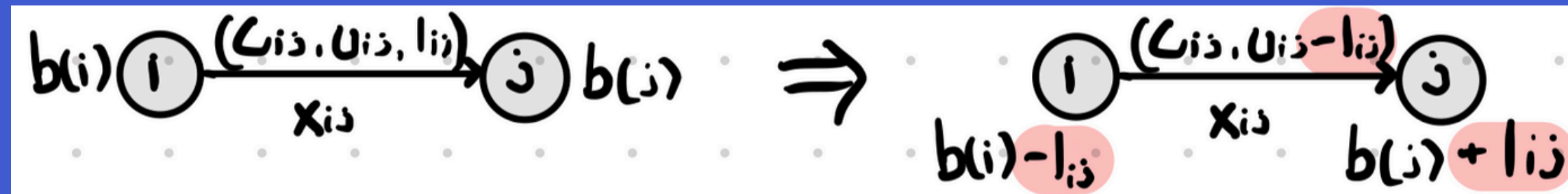
Para obtener la ruta mas corta entre 2 nodos se usa:

```
nx.shortest_path(G, source=nodo_origen, target=nodo_destino,  
weight='length')
```

En este caso el costo es la distancia.

# IMPORTANTE: COTAS MINIMAS

Las cotas minimas no existen en las librerias que utilizamos, con lo cual las debemos eliminar con la transformacion vista en clases.



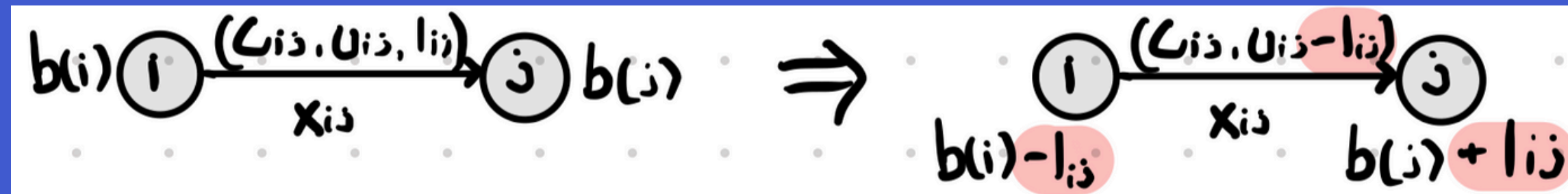
$$x_{ij} = x'_{ij} + l_{ij}$$



# IMPORTANTE: COTAS MINIMAS

Como incluir cotas minimas:

- Crear atributo de “cota\_min” para cada arco. Ejemplo:  
`H.add_edge("N2","N4",weight=12,cota_min=0,capacity=10)`
- Eliminar cotas minimas como fue visto en clases:



Nota: los signos en la transformación de las demandas van al revés.

- Resolver PPMC y reincorporar cotas minimas a los flujos con:

$$x_{ij} = x'_{ij} + l_{ij}$$

# IMPORTANTE: COTAS MINIMAS

Codigo para eliminar cotas minimas:

```
for u,v,a in H.edges(data=True):  
    if a["cota_min"] > 0:  
        nx.set_node_attributes(H,{u: {'demand': H.nodes[u]. ['demand']+a['cota_min']}})  
        nx.set_node_attributes(H,{v: {'demand': H.nodes[v]['demand']-a['cota_min']}})  
        nx.set_edge_attributes(H,{(u,v): {'capacity': a['capacity']-a['cota_min']}})
```

# IMPORTANTE: COTAS MINIMAS

Luego resolvemos el PPMC y reincorporamos las cotas minimas “ya enviadas”

```
result = nx.min_cost_flow(H, demand='demand', capacity='capacity', weight='weight')
for u,v,a in H.edges(data=True):
    result[u][v] = result[u][v] + a["cota_min"]
```

## TIP

**Recuerden que `nx.min_cost_flow()` entrega todos los flujos en el grafo trabajado, lo cual es un resultado muy importante en el desarrollo de este trabajo**



# RECOMENDACIONES T2

Incluyan sus nombres en el informe.

Entreguen todo en el formato solicitado.

Lean bien el enunciado.

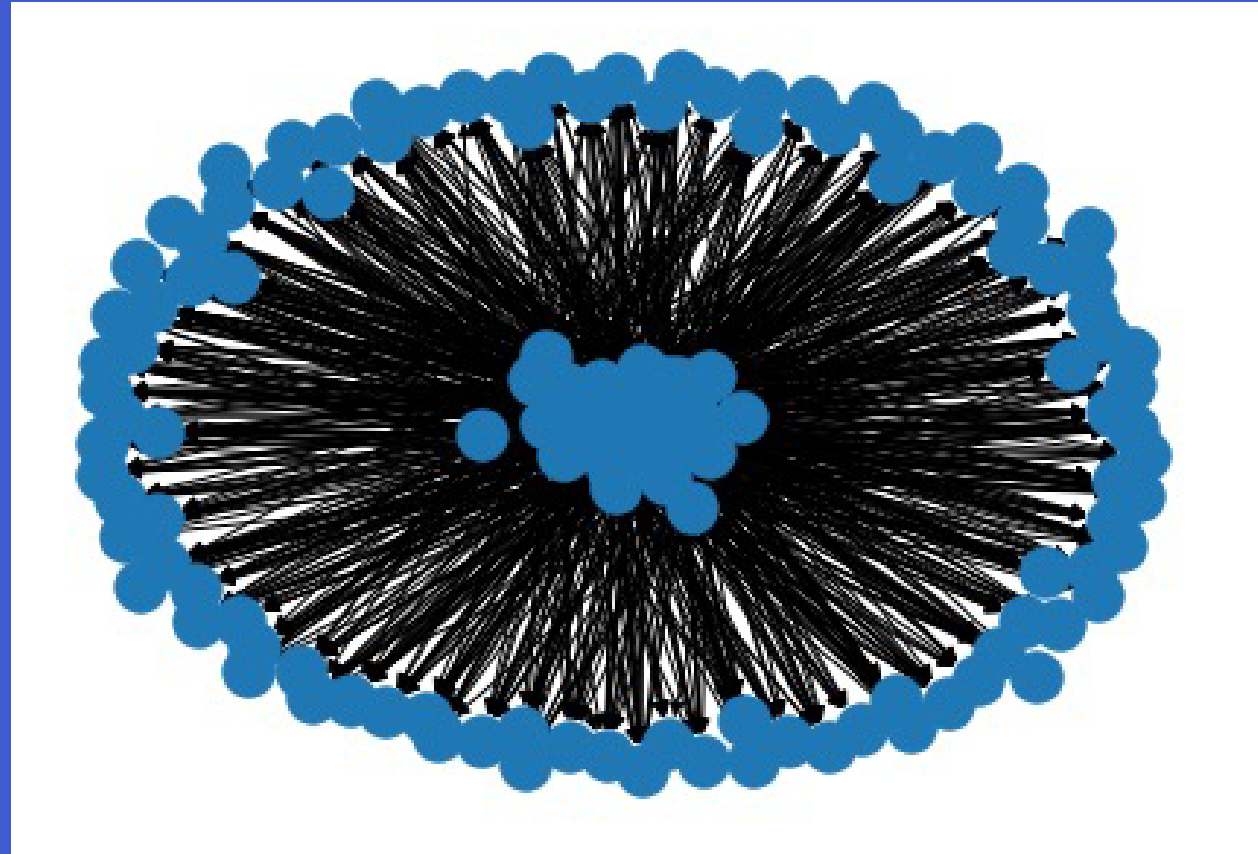
Si no logran resultados consistentes, expliquen qué hubieran esperado lograr y por qué.

Hagan todas las preguntas que necesiten (idealmente por el foro).

¡Empiecen con tiempo! Estamos con toda la disposición de ayudarles, pero no es posible si no inician a tiempo.

# RECOMENDACIONES T2

No entregar el siguiente tipo de grafo



# TALLER 2

**7 OCTUBRE 2024**

**MARTINA.BAQUEDANO@UC.CL**