Teoría de Grafos para Big Data

# Tarea

## Instrucciones

Esta tarea deben realizarla en grupos de 3 personas. Se espera que entreguen un documento en formato **PDF** con las respuestas a las 4 preguntas. Basta que uno de los integrantes entregue en la página del curso, pero tienen que **señalar el nombre de cada uno de los integrantes**. La fecha de entrega es para el domingo 15 de septiembre, antes de las 23:59. Todas las preguntas van a ser evaluadas con una nota entre 1 y 7. La calificación final es el promedio de las 4 preguntas.

## Pregunta 1: Page Rank

Explique con sus propias palabras el algoritmo de PageRank. En concreto se pide que respondan las siguientes preguntas:

1. En la fórmula:

$$PR_t(n_i) = \frac{1-d}{N} + d \sum_{n_j \in In(n_i)} \frac{PR_{t-1}(n_j)}{Out(n_j)}$$

   Que entrega el valor de PageRank para cada nodo en una iteración $t$, explique:

   - Qué representa el factor $d$ y por qué es importante en el cálculo de PageRank.
   - Cómo se refleja en la fórmula que un nodo va a ser más importante si recibe "links de nodos más importantes".

2. ¿Qué pasa en PageRank con los nodos desde los que salen muchas aristas?

3. En qué momento el algoritmo de PageRank para de hacer cálculos?

4. ¿Qué representa el PageRank final de cada nodo?

# Pregunta 2: RDF y SPARQL

Escriba en SPARQL las siguientes consultas al endpoint de Wikipedia. Recuerda que Wikidata tiene una lista de ejemplos de consultas en SPARQL y la lista completa de prefijos:

`https://www.wikidata.org/wiki/Wikidata:SPARQL\_query\_service/queries`
`https://en.wikibooks.org/wiki/SPARQL/Prefixes`

1. Los países cuyo presidente (o cabeza de estado) es mujer.

   Nótese que existen dos propiedades: P6 (*head of government*) y P35 (*head of state*). En muchos casos, como en Chile, estos dos son la misma persona. Pero en otros (por ejemplo Canadá), estos son personas distintas. Por ahora veamos solamente cabezas de estado, así que nos quedaremos con P35.

2. Las 20 cumbres más altas, con información de su cordillera (si existe). Ojo: las alturas pueden estar expresadas en unidades distintas. En general, un objeto de tipo `heightStatement` se relaciona mediante `psv:P2044` con una cantidad, y mediante `psn:P2044` con una cantidad normalizada.

3. Todos los escritores o escritoras chilenas vivos.
   **Hint**: usar `!bound()` después de haber ejecutado un OPTIONAL.

4. Las escritoras chilenas. Si han ganado un premio, inclúyalo.

Se espera que además de la consulta, incluya un *screenshot* con el resultado en el endpoint de Wikidata.

# Pregunta 3: Neo4J

Para realizar esta parte de la tarea, necesitarán iniciar el *sandbox* "Trumpworld" en la página de Neo4J. Se espera que para apoyar cada una de sus respuestas, adjunte un *screenshot* con el resultado de las consultas, en caso de corresponder.

## Estadísticas

Encuentre mediante una consulta en Cypher:

1. Los caminos más cortos entre Donald Trump y Vladimir Putin.

2. Las 5 personas con mayor grado entrante, saliente y bidireccional en el grafo. Escriba una consulta para cada tipo de grado.

3. Los 5 bancos (nodos de tipo Bank) más importantes en el grafo, y la gente conectada a ellas directamente o a través de una organización.

4. La gente conectada a través de una organización a Donald Trump.

### Pagerank

Como vimos en clase, podemos aplicar PageRank a cualquier grafo para obtener una métrica de importancia de los nodos en nuestra red. Neo4J trae implementado este algoritmo junto con otros.
Para esto, podemos ejecutar la siguiente consulta:

```
MATCH (c:Person)
WITH collect(c) AS people
CALL apoc.algo.pageRank(people) YIELD node, score
RETURN node.name AS name, score ORDER BY score DESC
```

1. ¿Quiénes son las personas más relevantes según Pagerank?
2. ¿Cómo se relacionan estas personas con las encontradas por grado?
3. Seleccione las 5 personas con mayor pagerank. ¿Qué nodos conectan a estas personas?

## Pregunta 4: GraphX y Pregel

Considere el archivo `ShortestPath.scala` publicado junto a esta tarea. Este archivo implementa el algoritmo de Dijkstra utilizando GraphX y Pregel. En esta pregunta se pide que expliquen con sus propias palabras cómo funciona el algoritmo. En concreto, se pide que respondan las siguientes preguntas:

1. ¿Cómo es el grafo que se está inicializando en la variable `graph`? ¿Qué diferencia tiene con el grafo que se inicializa en la variable `initialGraph`?
2. ¿Desde qué nodo está partiendo el algoritmo? ¿En qué variable se refleja?
3. ¿Cuál es el mensaje inicial que recibe la función `pregel`?
4. ¿Qué representa el valor enviado por los nodos en cada iteración? Esto se ve en la función `sendMsg`.
5. ¿Qué se hace cuando un nodo recibe múltiples mensajes?
6. ¿Qué hace la función `vprog` con todos los mensajes recibidos?

Además se debe explicar por qué este algoritmo corresponde efectivamente a calcular el *Shortest Path* desde un nodo inicial a todos los demás.