

# Programación Python para Big Data - Tarea lección 7

Kevin Martínez García

6 de julio de 2022

## Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Teoría de Grafos</b>	<b>2</b>

## 1. Introducción

En la actividad correspondiente a la lección 8 se nos pidieron dos tareas diferentes. En primer lugar, se pidió realizar una introducción a la *teoría de grafos* e indicar sus posibles usos en la tecnología relacionada con el Big Data. En segundo lugar, se pidió programar una aplicación que procesase datos en tiempo real haciendo uso de *Spark Streaming*.

## 2. Teoría de Grafos

La teoría de grafos es una de las ramas más modernas de las matemáticas. No fue hasta el año 1936 cuando apareció el primer texto que la desarrollaba como una teoría madura. Sin embargo, sus orígenes se remontan a los tiempos de Leonard Euler, quién resolvió el problema de los puentes de Königsberg [1].

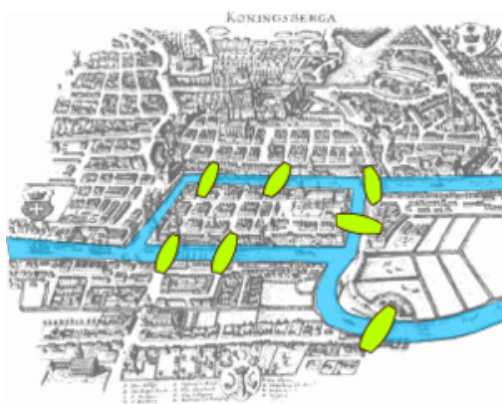


Figura 1: Los 7 puentes de Königsberg

Este problema plantea si es posible recorrer todas las regiones de la Figura 1 pasando una única vez por cada uno de los 7 puentes y volviendo al punto de partida. Euler resolvió este problema representando cada parte de tierra como un punto y cada puente como una arista que unía los puntos. De esta forma, el problema se reduce a comprobar si es posible recorrer el gráfico con un trazo continuo sin repetir líneas y volviendo al punto de partida (es decir, generar un ciclo sin aristas repetidas).

Visto el origen de esta teoría, podemos pasar a aspectos más formales y definir el concepto de grafo. Un grafo (no dirigido) es una terna  $G = (V, A, f)$  donde [2]:

- $V$  es un conjunto finito no vacío cuyos elementos se denominan vértices.
- $A$  es un conjunto finito cuyos elementos se denominan aristas
- $f$  es una asignación (aplicación de incidencia) que a cada arista de  $A$  le asigna un subconjunto de 1 ó 2 elementos. Es decir, sea  $e \in A$ ,  $f(e) = \{v_i, v_j\}$  o  $f(e) = \{v_i\}$  ( $v_i, v_j \in V$ ). A estos vértices se les llama extremos de la arista.

Gráficamente, un grafo no dirigido podría representarse como aparece en la Figura 2 a continuación.

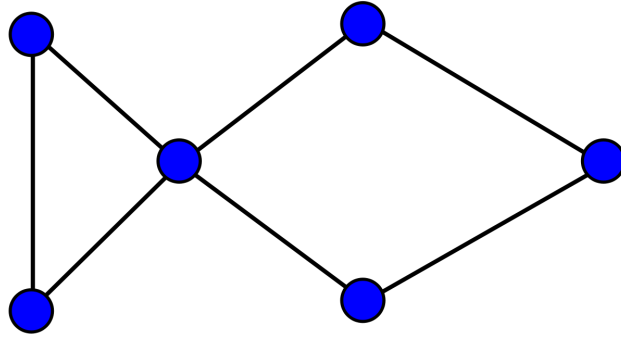


Figura 2: Grafo no dirigido

Esta definición vista puede extenderse para dar lugar a numerosas variaciones de los grafos. Por ejemplo, se podría añadir “orientación” a las aristas para dar lugar a los grafos dirigidos, o se podría asignar un peso a cada una de estas aristas, dando lugar a algoritmos tan importantes como el *algoritmo de Dijkstra* [3]. Los grafos también sirven como base de numerosos tipos abstractos de datos como los árboles o los montículos que también cuentan con numerosas aplicaciones en informática.

Cabe ahora preguntarse que posibles aplicaciones podría tener la teoría de gráficos en el Big Data. El uso de grafos para relacionar cantidades masivas de datos surge a iniciativa de *Google* que buscaba una forma efectiva de relacionar la inmensidad de documentos presentes en la Web. Con el uso de grafos, Google consiguió desarrollar un motor de búsqueda extremadamente rápido y potente, mucho más avanzado que los de sus competidores.

A nivel práctico, cuando accedemos a una Web que contiene un hipervínculo a otra web podemos entender ambas webs como nodos de un grafo, y dicho hipervínculo como una arista que une ambos nodos. Con esta intuición, es fácil ver que la Web puede modelarse como un grafo dirigido con una cantidad creciente de nodos. Por supuesto, la tarea de modelar la Web como un grafo es complicada y entran en juego muchas cuestiones como la “calidad” de los enlaces presentes en *Google*, filtrado de enlaces maliciosos, eliminación y filtrado de resultados idénticos o las famosas *Google Bombs* [4]. Existen además otros proyectos como *Neo4j* [5] que consiste en una implementación muy eficiente de bases de datos orientadas a grafos, que son simplemente bases de datos que utilizan la abstracción de los grafos para almacenar y devolver información [6].

En resumen, la teoría de grafos permite representar y relacionar de forma efectiva colecciones enormes de datos y por tanto, resulta esencial para el desarrollo del Big Data. Tanto es así que el desarrollo del Big Data está haciendo progresar a pasos agigantados esta rama de las matemáticas, como se indica en el artículo referenciado [7].

## Referencias

- [1] de, C. (2003). problema matemático. [online] Wikipedia.org. Available at: [https://es.wikipedia.org/wiki/Problema\\_de\\_los\\_puentes\\_de\\_K%C3%B6nigsberg](https://es.wikipedia.org/wiki/Problema_de_los_puentes_de_K%C3%B6nigsberg) [Accessed 20 Jun. 2022].
- [2] Wikipedia. (2021). Grafo no dirigido. [online] Available at: [https://es.wikipedia.org/wiki/Grafo\\_no\\_dirigido#:~:text=Un%20grafo%20no%20dirigido%20es](https://es.wikipedia.org/wiki/Grafo_no_dirigido#:~:text=Un%20grafo%20no%20dirigido%20es) [Accessed 20 Jun. 2022].

- [3] Wikipedia. (2021). Algoritmo de Dijkstra. [online] Available at: [https://es.wikipedia.org/wiki/Algoritmo\\_de\\_Dijkstra](https://es.wikipedia.org/wiki/Algoritmo_de_Dijkstra).
- [4] Wikipedia. (2022). Google bomb. [online] Available at: [https://es.wikipedia.org/wiki/Google\\_bomb](https://es.wikipedia.org/wiki/Google_bomb) [Accessed 20 Jun. 2022].
- [5] Neo4j Graph Database Platform. (2017). Neo4j Graph Platform – The Leader in Graph Databases. [online] Available at: <https://neo4j.com/>.
- [6] Wikipedia Contributors (2019). Graph database. [online] Wikipedia. Available at: [https://en.wikipedia.org/wiki/Graph\\_database](https://en.wikipedia.org/wiki/Graph_database).
- [7] Ornes, S. (2021). How Big Data Carried Graph Theory Into New Dimensions. [online] Quanta Magazine. Available at: <https://www.quantamagazine.org/how-big-data-carried-graph-theory-into-new-dimensions-20210819/>.