

Propuesta de una Herramienta Web para el Aprendizaje de Grafos: Aplicación del Algoritmo de PageRank

Federico Nahuel Mamani¹

Adair Martins¹

Departamento de Computación Aplicada

¹Facultad de Informática, Universidad Nacional del Comahue, Neuquén, Argentina

nahuel.mamani@est.fi.uncoma.edu.ar, adair.martins@fi.uncoma.edu.ar

Resumen

El gran avance que las nuevas Tecnologías de la Información y Comunicación (TIC) han tenido en las últimas décadas, ha impulsado en forma significativa la utilización de herramientas informáticas en la resolución de problemas de distintos campos del conocimiento. Su integración en el ámbito educativo es un factor clave para la mejora y fortalecimiento de los procesos de enseñanza y aprendizaje. La teoría de grafos forma parte de diversos estudios de matemática discreta, investigación operativa, diseño de redes, entre otros, y tiene innumerables aplicaciones en las ciencias de la computación, biología, sistemas sociales o físicos y en diversas especialidades de ingeniería. En este contexto se presenta una propuesta para el desarrollo de una herramienta web para ser utilizada en el aprendizaje de algoritmos de grafos, principalmente para las carreras dictadas en la Facultad de Informática (FAIF). Está enmarcada en la línea “Desarrollo y Uso de Recursos TIC” del Proyecto de Investigación: “Computación Aplicada a las Ciencias y Educación”, FAIF, Universidad Nacional del Comahue. Para el desarrollo e implementación se empleó la arquitectura cliente-servidor y diferentes tecnologías web open source. Para mostrar sus funcionalidades se presenta un caso de aplicación del algoritmo de PageRank.

Palabras Clave: Herramienta web, TIC, Grafos, Algoritmos, PageRank, Enseñanza y Aprendizaje

1. Introducción

En el ámbito de las Ciencias de la Computación, uno de los conceptos matemáticos de relevancia es el de grafo. Esto se refleja en la existencia de una rama de las matemáticas dedicada exclusivamente al estudio de los grafos, conocida como Teoría de Grafos [1], originada en el siglo XVIII, y que está presente en las asignaturas de todas las carreras relacionadas con la ciencia computacional. En este sentido, en varias materias dictadas en las carreras de la Facultad de Informática se incluyen contenidos de algoritmos sobre grafos, por lo tanto fortalecer su aprendizaje a través de la utilización de herramientas computacionales es de suma importancia dado que permitiría mejorar la comprensión por parte de los estudiantes de los conceptos teóricos y prácticos, facilitando el análisis y la resolución de problemas.

Se puede mencionar también que en los últimos años han surgido diferentes propuestas para su incorporación en los planes de estudio de los niveles de educación primario y secundario [2],[3],[4].

Básicamente, un grafo es una estructura que consiste de un conjunto de nodos interconectados mediante arcos, y permite modelar diversas situaciones donde se

requiere representar relaciones entre un conjunto de elementos. Según [5], “los grafos constituyen una buena herramienta para conceptualizar situaciones, extraer pautas y, de forma mucho más evidente, para llegar a entender esquemas y transferirlos a situaciones nuevas”

Estas estructuras resultan de gran utilidad dado que es posible aplicar sobre las mismas una amplia variedad de *algoritmos para grafos*, y cuyos resultados pueden contribuir a facilitar el análisis y la resolución de distintas problemáticas asociadas a las situaciones que modelan. Tienen aplicación en diferentes ramas científicas, inclusive en situaciones de la vida cotidiana. Por ejemplo:

- En bioquímica, para representar rutas metabólicas, y a partir de ellos encontrar todas las posibles secuencias de reacciones entre dos compuestos químicos [6].
- En análisis de redes sociales [7], para el análisis e investigación de las estructuras sociales por medio de estas redes y de la Teoría de Grafos.
- Para modelar redes de autopistas y determinar con mayor facilidad la ruta de menor longitud entre dos locaciones distintas.

Los grafos y sus algoritmos son de naturaleza visual y dinámica, motivo por el cual un entorno gráfico y dinámico proporciona una gran ayuda en la comprensión del funcionamiento de estos algoritmos [8]. Además es deseable que este entorno sea interactivo y favorezca la experimentación, para lograr por medio de la misma la consolidación del aprendizaje. En este sentido, las herramientas TIC resultan ser muy propicias para cumplir tales finalidades y requerimientos. En la actualidad, cada vez es mayor la adopción de las mismas en varias áreas del conocimiento, sobre todo en educación y ciencias [9].

Del relevamiento de distintas herramientas informáticas para el aprendizaje de los algoritmos de grafos, se puede nombrar a Grafos [10], ALGraf [11], IAGraph [12], VisuAlgo [13], Data Structure Visualizations [14] y Algorithm Visualizer [15]. Si bien todas permiten la visualización de grafos y la aplicación de algoritmos sobre grafos definidos por el usuario, presentan diferencias respecto a determinadas características que son consideradas relevantes para mejorar el aprendizaje de estos algoritmos. En cuanto a la forma en que el usuario ingresa sus grafos a la aplicación, Grafos, ALGraf, IAGraph y VisuAlgo disponen de un editor visual de grafos, lo que puede ser de gran ayuda sobre todo para usuarios novatos al posibilitar un modelado más intuitivo y fácil. Sin embargo, requieren de instalación previa de ciertos programas o paquetes para su funcionamiento. Es deseable que toda herramienta orientada a educación sea de fácil acceso y puesta en marcha, como así también utilizable en distintas plataformas. VisuAlgo, Algorithm Visualizer y Data Structure Visualizations cumplen con tales requerimientos, dado que son aplicaciones web que para su uso solo requieren contar con algún navegador web. Estas tres últimas también se destacan por permitir la visualización de la ejecución de los algoritmos y sus resultados que facilita sustancialmente la comprensión de su funcionamiento y el análisis de los valores obtenidos. Algorithm Visualizer, a diferencia de las herramientas restantes, no limita al usuario a experimentar con un conjunto limitado de algoritmos ya que posee un editor de código fuente que permite al usuario escribir sus propios algoritmos.

En el presente trabajo se presenta el caso de aplicación de PageRank [16],[17], un algoritmo creado por Google para establecer un ranking de importancia de las páginas web en base a la estructura de grafo de la web. Dicho caso de aplicación se desarrolla en el contexto de una herramienta en la que se está trabajando actualmente en la línea de investigación. Consiste en una aplicación web

basada en una arquitectura cliente-servidor, y puede ser utilizada desde cualquier navegador web moderno. Está compuesta de un editor visual de grafos y de un editor de código fuente con el que es posible escribir algoritmos que se ejecutarán sobre el grafo modelado, y cuyos resultados pueden ser visualizados sobre el mismo. La herramienta otorga al usuario cierta flexibilidad al darle la libertad de elegir el lenguaje de programación con el cual escribir sus algoritmos. Así también, dispondrá de un conjunto de algoritmos ya implementados y listos para su ejecución, que estará integrado por algunos de los más conocidos y/o de mayor aplicación. La finalidad de la herramienta propuesta es apoyar y fortalecer el aprendizaje del funcionamiento de los algoritmos sobre grafos, y está dirigida principalmente a estudiantes de las carreras de Ciencias de la Computación.

2. Marco Teórico

2.1 Definición formal de Grafo

Un *grafo* es un par $G = (N, A, g)$ donde N es un conjunto finito no vacío de elementos denominados *nodos* (vértices), A es un conjunto de *arcos* y g es un función que asocia a cada arco a perteneciente a A con un par no ordenado (x, y) , siendo x e y nodos pertenecientes a N . Se dice que a es un arco con vértices extremos x e y [18].

2.2 Definición formal de Grafo Dirigido

Un *grafo dirigido* es un par $G = (N, A, g)$ donde N es un conjunto finito no vacío de elementos denominados *nodos* (vértices), A es un conjunto de *arcos* y g es un función que asocia a cada arco a perteneciente a A con un par ordenado (x, y) , siendo x e y nodos pertenecientes a N . Se dice que a es un arco con vértice inicial x y vértice final y [18].

2.3 PageRank

PageRank (PR) es un método que fue implementado a través de un algoritmo originalmente utilizado por Google que asigna a cada página web de un conjunto dado, un puntaje que refleja su importancia dentro del conjunto. A este puntaje se lo denomina *valor de PageRank*. Ante una consulta, el buscador utiliza estos puntajes para determinar el nivel de relevancia de las páginas, y retorna en primer lugar aquellas con un puntaje más alto. Para calcular los puntajes, PageRank utiliza la estructura de enlaces de la web [17]. Tal como se enuncia en [16], una página web tiene un valor de PageRank alto si es apuntada por muchas otras páginas, o bien si es apuntada por páginas con puntajes altos.

PageRank tiene una base intuitiva en el concepto de *random walks* sobre grafos [19]: suponga que un navegante aleatorio empieza a navegar la web desde una página cualquiera. El navegante puede hacer clic en forma aleatoria sobre alguno de los enlaces presentes en la página en la que se encuentra actualmente con una probabilidad d a la que se denomina *damping factor*, o bien con probabilidad $1-d$ accede aleatoriamente a cualquier otra página web. Este proceso se repite indefinidamente. Luego, el valor de PageRank de una página P puede ser interpretado como la probabilidad de que el navegante aleatorio se encuentre en P al finalizar el proceso.

Formalmente, PageRank es definido en [20] tal como se realiza a continuación. Sean q_i el número de enlaces salientes que posee la página i , n el número total de páginas web, d el *damping factor* que por lo general adquiere el valor 0.85, π un vector columna denominado *vector PageRank*, y $H = (h_{i,j})$ una matriz cuadrada de tamaño n tal que $h_{i,j} = 1/q_i$ si existe un enlace desde la página i a la página j , y $h_{i,j} = 0$ en caso contrario. El valor $h_{i,j}$ corresponde a la probabilidad de acceder a la página j desde la página i en un

paso, a partir de hacer clic en alguno de los enlaces que aparecen en esta última. El valor de PageRank correspondiente a la página j es π_j , y se define recursivamente como se muestra en la ecuación (1). Para su mejor comprensión, se recomienda consultar en [21], donde se explican detalladamente los componentes de dicha ecuación.

$$\pi_j = \frac{1-d}{n} + d \sum_{i=1}^n \pi_i h_{i,j} \quad (1)$$

El vector PageRank se define mediante la ecuación (2), donde I es un vector columna con todas sus posiciones con valor 1. El vector PageRank conforma una distribución de probabilidad sobre las páginas web.

$$\pi = \frac{1-d}{n} \cdot I + d \cdot H^T \cdot \pi \quad (2)$$

El vector PageRank puede ser computado en forma iterativa utilizando la ecuación (3), donde $\pi^{(k)}$ denota al vector PageRank obtenido tras llevarse a cabo k iteraciones.

$$\pi^{(t)} = \frac{1-d}{n} \cdot I + d \cdot H^T \cdot \pi^{(t-1)} \quad (3)$$

El cómputo se inicia a partir de un vector arbitrario $\pi^{(0)}$ (generalmente con todas sus posiciones inicializadas con el valor $1/n$) y se detiene cuando se alcanza un criterio de convergencia preestablecido.

Si H es una matriz estocástica derecha [22] (en [20] se explica como transformar a H en una matriz de este tipo), la ecuación (2) resulta equivalente a la expresión que se muestra en la ecuación (4), donde $G = (\frac{1-d}{n} \cdot E + d \cdot H)$, y E es una matriz cuadrada de tamaño n , con todas sus posiciones con valor 1.

$$\pi = G \cdot \pi \quad (4)$$

Resolver la ecuación (4) equivale a encontrar el autovector π para G con autovalor 1 [23]. Para este caso en particular, aplica el Teorema de Perrón-Frobenius [24], por lo que el vector PageRank existe y es único, pudiendo ser calculado en forma iterativa aplicando el Método de las Potencias [25].

Si se considera a las páginas web como nodos y a los links entre ellas como arcos, resulta que el método de PageRank es aplicable a grafos dirigidos en general. A continuación se listan algunos casos concretos de aplicación:

- En [26] se explica el uso de PageRank para establecer una función de costo que asigna un puntaje a cada calle de una determinada zona. A partir de esta función se diseñó un mecanismo de aprendizaje por refuerzo [27] que permite determinar las decisiones más óptimas para el control de los semáforos de dicha zona, y así evitar el congestionamiento en el tráfico.
- En [28] se presenta una aplicación de PageRank para medir la calidad e influencia de los artículos científicos, basándose en la estructura de red que conforman las citaciones entre artículos, como así también en la influencia personal de los autores y en las fechas de publicación.
- Se utiliza PageRank para medir el grado de autoridad y/o influencia de los usuarios en una red social. En [29] se propone una variación del algoritmo para esta finalidad.

3. Diseño e implementación de la herramienta

Se presenta la propuesta de una herramienta web para apoyar el aprendizaje de algoritmos de grafos. Sus funcionalidades básicas son listadas a continuación:

- Edición visual de grafos
- Edición de código fuente con la posibilidad de seleccionar el lenguaje de programación a utilizar. El usuario puede escribir sus propios algoritmos, como así también modificar las implementaciones de algunos de los algoritmos más

conocidos que son provistas por la herramienta.

- Aplicación de algoritmos sobre el modelo creado por el usuario.
- Visualización de los resultados intermedios y finales obtenidos de la ejecución del algoritmo, con la posibilidad de observarlos representados directamente sobre el modelo gráfico. Con esto se busca facilitar el análisis de los resultados.

La herramienta consta de una arquitectura del tipo cliente-servidor, cuya disposición inicial puede observarse en la figura 1. El lado cliente se compone de tres módulos principales, cada uno de ellos vinculado con una funcionalidad bien definida: visualización y manipulación del grafo creado por el usuario, edición de código fuente y gestión de conexiones con el servidor. Cuando el cliente emite una petición de ejecución al servidor, en la misma se adjuntan una representación textual de grafo modelado, junto con el código fuente que aparece en el editor en ese momento. Este código fuente puede ser de creación propia del usuario, o bien alguna implementación que la herramienta provee. En la petición también se indica el lenguaje en el que está escrito el código fuente.

Por otra parte, el servidor consta de un módulo para el manejo de las peticiones entrantes, y de un módulo por cada lenguaje de programación admitido por la herramienta. Mediante esta disposición, el servidor recibe la petición proveniente del cliente, verifica el lenguaje indicado en la misma e invoca al módulo correspondiente para la ejecución del código fuente incluido en la petición. El resultado obtenido es retornado hacia el cliente para su visualización.

En lo que concierne a la implementación, se encuentra en construcción un prototipo de la herramienta empleando diferentes tecnologías web open source [30]. Hasta el momento, el proceso de desarrollo se ha concentrado

mayoritariamente en el lado cliente, destacándose aquí el uso de librerías de Javascript tales como SigmaJS [31] y Codemirror [32]. SigmaJS permite la visualización de grafos y la interacción de los usuarios con los mismos. Por su parte, Codemirror provee de un componente de edición de código fuente para navegadores web, incluyendo varios modos para programar en diferentes lenguajes.

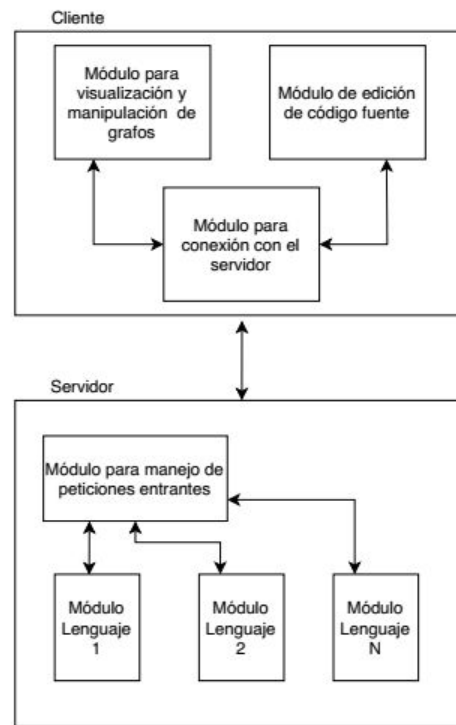


Figura 1: Arquitectura inicial de la herramienta propuesta

Se espera que la herramienta propuesta facilite la comprensión del funcionamiento de los algoritmos de grafos por medio de la experimentación. Esto implica ejecutar los mismos utilizando distintos grafos y valores de parámetros de entrada, e inclusive realizar modificaciones en su código fuente, para posteriormente llevar a cabo un análisis y comparación entre los resultados obtenidos que conlleve a un mejor entendimiento de los algoritmos trabajados.

4. PageRank: un caso de aplicación de la herramienta

A continuación se lleva a cabo una demostración de una ejecución de PageRank en el prototipo de la herramienta.

En la figura 2 se muestra la pantalla principal, compuesta de un menú en la parte superior, el editor visual de grafos y el editor de código fuente. Las opciones incluidas en el menú permiten:

- Ejecutar acciones y establecer configuraciones sobre ambos editores, como por ejemplo: elegir si el grafo a modelar será de tipo dirigido, determinar el lenguaje de programación a utilizar en el editor de código fuente, entre otras.
- Seleccionar alguno de los algoritmos provistos, cuya código aparecerá en el editor correspondiente, en el lenguaje indicado por el usuario.
- Ejecutar la implementación que se encuentra en el editor de código.

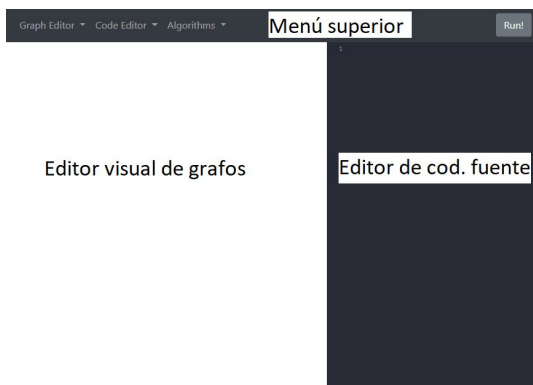


Figura 2: Pantalla principal

En la figura 3 se muestra la pantalla luego de seleccionarse PageRank. En este ejemplo se utiliza la implementación del algoritmo en Python, originalmente provista por Networkx [33],[34], un paquete de dicho lenguaje para la creación, manipulación y estudio de la estructura, dinámicas y funciones de grafos complejos.

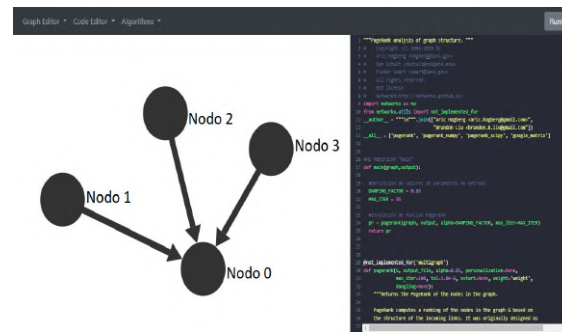


Figura 3: Vista de la pantalla previamente a ejecutarse PageRank

Al presionar el botón “Run”, se ejecuta en el servidor el código que se halla en el editor sobre el grafo construido. Tras la finalización, se despliega un widget con los resultados obtenidos. La figura 4 corresponde al widget exhibido luego de haberse aplicado PageRank sobre el grafo de la figura 3. En base a lo explicado en la sección 2.3, se justifica el hecho de que el Nodo 0 es el de valor de PageRank más alto ya que posee la mayor cantidad de arcos entrantes. Por lo tanto, se dice que el Nodo 0 es el de mayor relevancia en el grafo de la figura 2.

Results		
#	Node Label	Pagerank value
1	Node 0	0.5419853569696174
2	Node 1	0.1526715476767942
3	Node 2	0.1526715476767942
4	Node 3	0.1526715476767942

See results in graph

Figura 4: Widget desplegado al finalizar la ejecución de PageRank

El widget dispone de una opción “See results in graph” para visualizar los resultados sobre el modelo gráfico, mostrándose los nodos de mayor valor de PageRank con un mayor

tamaño respecto a los nodos con valores inferiores. Para este ejemplo, el nodo de mayor valor es el Nodo 0, por lo que aparece como el de mayor dimensión, tal como se puede ver en la figura 5.

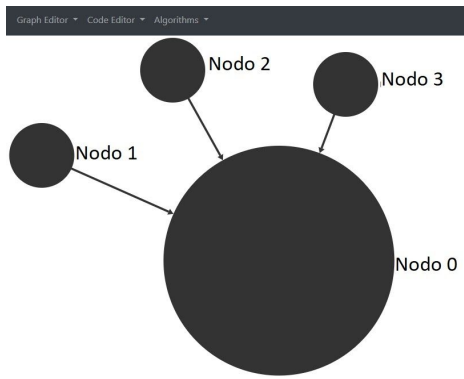


Figura 5: Resultados de la ejecución de PageRank visualizados en el grafo

También se pueden consultar los valores de los resultados intermedios, por el momento mediante acceso a la consola web del navegador. En la figura 6 se exhibe una vista de la consola web con los valores de PageRank intermedios obtenidos durante la ejecución anterior.

```

Node 3: 0.15267255874046465

Pagerank values in iteration 27:
Node 0: 0.5419862685911386
Node 1: 0.1526712438029538
Node 2: 0.1526712438029538
Node 3: 0.1526712438029538

Pagerank values in iteration 28:
Node 0: 0.5419837537731491
Node 1: 0.15267208207561694
Node 2: 0.15267208207561694
Node 3: 0.15267208207561694

```

Figura 6: Resultados intermedios de la ejecución de PageRank

5. Conclusiones y trabajos futuros

Se ha presentado una propuesta de una herramienta web cuya finalidad es mejorar la comprensión por parte de los estudiantes del funcionamiento de los algoritmos sobre grafos. La misma se compone de un editor visual de grafos y de un editor de código fuente con el que es posible escribir algoritmos que se ejecutarán sobre el grafo modelado, y cuyos resultados pueden ser visualizados sobre el mismo.

Se llevó a cabo una descripción de su arquitectura, basada en el modelo cliente-servidor. En lo que concierne a su implementación, se encuentra en proceso de construcción un prototipo de la herramienta, utilizando tecnologías web open source. Para mostrar sus funcionalidades principales, se desarrolló un caso de aplicación del algoritmo de PageRank.

Actualmente se continúa trabajando en el prototipo, con foco en el desarrollo del lado servidor. Se pretende continuar avanzando con el estudio en esta línea de investigación mediante la incorporación de otros algoritmos de grafos y de nuevas características, entre las que se destaca la inclusión de directivas que el usuario podrá usar en sus programas para exhibir animaciones que permitan visualizar las ejecuciones (por ejemplo: coloreado de arcos/nodos y resaltado de la línea del código en ejecución).

Se llevarán a cabo evaluaciones del prototipo mediante la realización de encuestas a alumnos de la carrera de Licenciatura en Ciencias de la Computación de la Facultad de Informática, esperando poder validar la herramienta. Las mismas estarán orientadas a determinar la opinión de los usuarios respecto a cuestiones tales como usabilidad, navegabilidad y facilidad de comprensión de ciertos comportamientos de los algoritmos al ser ejecutados a partir de un conjunto de casos propuestos. También se busca detectar posibles fuentes de mejoras con el objetivo de seguir facilitando y fortaleciendo los procesos

de enseñanza y aprendizaje.

Referencias Bibliográficas

- [1] E. A. Maziarz, “Graph Theory: 1736-1936. N. L. Biggs , E. K. Lloyd , R. J. Wilson”, *Isis*, vol. 70, núm. 1. pp. 164–165, 1979, doi: 10.1086/352170.
- [2] H. S. Bueno, “TFM: Diseño de una unidad didáctica sobre grafos en 2º de Bachillerato”, Universitat Jaume I, 2018.
- [3] D. Ferrarello y M. F. Mammana, “Graph Theory in Primary, Middle, and High School”, *ICME-13 Monographs*. pp. 183–200, 2018, doi: 10.1007/978-3-319-70308-4_12.
- [4] D. Lessner, “Graph Theory in High School Education”, Department of Software and Computer Science Education, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic, 2011.
- [5] M. C. Benarroch, *Nudos y nexos : redes en la escuela*. Editorial Síntesis, 1989.
- [6] Z. Abd Algfoor, M. Shahrizal Sunar, A. Abdullah, y H. Kolivand, “Identification of metabolic pathways using pathfinding approaches: a systematic review”, *Brief. Funct. Genomics*, vol. 16, núm. 2, pp. 87–98, mar. 2017.
- [7] M. A. Sánchez, “Visualización de datos utilizando grafos: ejemplos de aplicación”, *Revista “CEA”*, vol. 1, núm. 1, 2017.
- [8] M. G. S. Torrubia y M. Gloria Sánchez Torrubia, “Especificaciones eMathTeacher, creación del Modelo Granular Lingüístico de la evaluación del aprendizaje humano, y diseño de sistemas que implementan estos conceptos”. doi: 10.20868/upm.thesis.39386.
- [9] C. Allan, S. Parra, y A. Martins, “Objetos de aprendizaje para la interpretación geométrica de métodos numéricos: uso de GeoGebra”, *XI Congreso de Tecnología en Educación y Educación en Tecnología (TE&ET 2016)*, pp. 541–546, 2016.
- [10] A. Villalobos, “Grafos : herramienta informática para el aprendizaje y resolución de problemas reales de teoría de grafos”, en *X Congreso de Ingeniería de Organización*, Valencia, 2006.
- [11] “ALGraf”, *Departamento de Matemática Aplicada a las Tecnologías de la Información y las Comunicaciones*. [En línea]. Disponible en: <http://www.dma.fi.upm.es/personal/gregorio/grafos/algraf/>. [Consultado: 11-abr-2020].
- [12] J. C. D. Núñez, “Aplicación Integral de Grafos: Búsquedas”, Universidad Politécnica de Madrid, 2012.
- [13] “VisuAlgo - Visualización de estructuras de datos y algoritmos mediante animación (Spanish)
”. [En línea]. Disponible en: <https://visualgo.net/es>. [Consultado: 29-abr-2020].
- [14] “Data Structure Visualization”. [En línea]. Disponible en: <https://www.cs.usfca.edu/~galles/visualization/Algorithms.html>. [Consultado: 29-abr-2020].
- [15] “Algorithm Visualizer”, *Algorithm Visualizer*. [En línea]. Disponible en: <https://algorithm-visualizer.org/>. [Consultado: 29-abr-2020].
- [16] L. Page, S. Brin, R. Motwani, y T. Winograd, “The PageRank citation ranking: Bringing order to the Web”, en *7th International World Wide Web Conference*, Brisbane, Australia, 1998, pp. 161–172.
- [17] S. Brin y L. Page, “The anatomy of a large-scale hypertextual Web search engine”, *Computer Networks and ISDN Systems*, vol. 30, núm. 1–7. pp. 107–117, 1998, doi: 10.1016/s0169-7552(98)00110-x.

- [18] J. L. Gersting, *Mathematical Structures for Computer Science*. Macmillan, 2007.
- [19] G. Grimmett, “Random walks on graphs”, *Probability on Graphs*. pp. 1–20, doi: 10.1017/cbo9780511762550.002.
- [20] M. Franceschet, “PageRank: standing on the shoulders of giants”, *Communications of the ACM*, vol. 54, núm. 6. pp. 92–101, 2011, doi: 10.1145/1953122.1953146.
- [21] J. Lin y C. Dyer, “Data-intensive text processing with MapReduce”, *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Tutorial Abstracts on - NAACL '09*. 2009, doi: 10.3115/1620950.1620951.
- [22] F. E. Szabo, “S”, en *The Linear Algebra Survival Guide*, F. E. Szabo, Ed. Boston: Academic Press, 2015, pp. 320–377.
- [23] J. Liesen y V. Mehrmann, *Linear Algebra*. Springer, 2015.
- [24] “The Perron-Frobenius theorem”, *Interaction Models*. pp. 97–99, 1977, doi: 10.1017/cbo9780511662126.007.
- [25] R. L. Burden y J. Douglas Faires, *Numerical Analysis*. Cengage Learning, 2011.
- [26] F. Pop y C. Dobre, “An Efficient PageRank Approach for Urban Traffic Optimization”, *Mathematical Problems in Engineering*, vol. 2012. pp. 1–9, 2012, doi: 10.1155/2012/465613.
- [27] A. Nandy y M. Biswas, “Reinforcement Learning Basics”, *Reinforcement Learning*. pp. 1–18, 2018, doi: 10.1007/978-1-4842-3285-9_1.
- [28] C. Ji, Y. Tang, y G. Chen, “Analyzing the Influence of Academic Papers Based on Improved PageRank”, *Emerging Technologies for Education*. pp. 214–225, 2020, doi: 10.1007/978-3-030-38778-5_24.
- [29] H. Zhao, X. Xu, Y. Song, D. L. Lee, Z. Chen, y H. Gao, “Ranking Users in Social Networks with Motif-based PageRank”, *IEEE Trans. Knowl. Data Eng.*, pp. 1–1, 2019.
- [30] P. Kavanagh, “Open Source Software”, *Open Source Software*. pp. 1–17, 2004, doi: 10.1016/b978-155558320-0/50002-7.
- [31] “Sigma js”. [En línea]. Disponible en: <http://sigmajs.org/>. [Consultado: 11-abr-2020].
- [32] “CodeMirror”. [En línea]. Disponible en: <https://codemirror.net/>. [Consultado: 23-abr-2020].
- [33] “NetworkX — NetworkX”. [En línea]. Disponible en: <https://networkx.github.io/>. [Consultado: 11-abr-2020].
- [34] “pagerank — NetworkX 1.10 documentation”. [En línea]. Disponible en: https://networkx.github.io/documentation/networkx-1.10/reference/generated/networkx.algorithms.link_analysis.pagerank_alg.pagerank.html. [Consultado: 13-abr-2020].