UNIVERSIDAD NACIONAL DE LA AMAZONÍA PERUANA

FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMATICA

TESIS PARA OPTAR EL TÍTULO
DE
INGENIERO DE SISTEMAS E INFORMATICA

"APROXIMACION MATEMATICA Y COMPUTACIONAL DEL MOTOR DE BUSQUEDA GOOGLE"

AUTOR : BACH. ROJAS TENAZOA JULIO MARTIN

ASESOR : ING. CARLOS ALBERTO GARCIA CORTEGANO

IQUITOS - PERÚ
2016

AGRADECIMIENTOS

Primeramente, debo agradecer a mi asesor, Ing. Msc. **Carlos Alberto García Cortegano**, con sus argumentaciones científicas y sus sugerencias, tuve la oportunidad de enriquecer mis conocimientos.

Agradezco a mis familiares que siempre creyeron en mí, en especial para mi madre, **Doris** y mi padre **Julio** que hace poco partió al cielo para estar con Dios, que me dieron no solamente la vida, también educación y condiciones de estudio y que junto a mis hermanos **Betsi**, **Cristian** y a mi pareja **Marizabeth**, de diferentes maneras me ayudaron mucho a concluir esta etapa de mi vida.

Debo de agradecer al Ing. Msc. **Fernando Javier Salas Barrera** y al Lic. Mg. **Manuel Tuesta Moreno**, por su amistad y apoyo en este trabajo de investigación.

No puedo olvidar de una persona muy especial, a quien siempre recurro en momentos difíciles, mi ángel de la guarda y hermana **Karen Milagros**, siempre presente.

Mi agradecimiento a la Facultad de Ingeniería de Sistemas e Informática, que me posibilito la realización de este trabajo.

.

A mis hijas Doris y Bianca

RESUMEN

En el presente proyecto de tesis se presenta al algoritmo de PageRank de forma detallada retomando la teoría de cadena de Markov, contemplando algunas definiciones y propiedades útiles para la compresión del algoritmo, bien como el método de las potencias para la obtención de los valores propios y vectores propios dominantes de una matriz estocástica, así como el teorema de Perron-Frobenius que bajo ciertas condiciones nos asegura la existencia del vector buscado(vector PageRank); con el objetivo de proveer un principio teórico para el entendimiento del algoritmo. Enseguida, a fin de entender las diferentes situaciones que el modelo puede enfrentar, una simulación será presentada en este trabajo. En particular estamos interesados en construir de manera aproximada un buscador tipo Google. En el primer capítulo de esta tesis enunciaremos el cimiento matemático necesario para el entendimiento del algoritmo, explicaremos Cadena de Markov, Norma de vectores, Dependencia e independencia lineal, valores y vectores propios, multiplicidad algebraica y geométrica, matriz diagonizable, teorema de Perron-Frobenius y métodos de las potencias; seguidamente detallaremos definiciones de Web Crawler así como el funcionamiento del mismo, haciendo uso del software OpenWebSpider, luego abordaremos el software PAJEK, que nos permite visualizar nuestro espacio web analizado por el Crawler y finalmente daremos un detalle breve del uso del Matlab, en particular orientado al algebra lineal. En el segundo capítulo detallaremos el algoritmo de ordenación usado por Google y lo aplicaremos matemáticamente, haciendo un planteamiento del modelo. En el tercer capítulo, mediante dos casos de estudios y con dos grafos asociados con una red que consta de 5 y 4 páginas, detallaremos el algoritmo de ordenamiento de Google, empleando las definiciones vistas en los capítulos preliminares. La última parte estará dedicada a la simulación del algoritmo, para ello detallaremos las acciones que realizará el Crawler, seguidamente mediante la integración entre PAJEK y Matlab generemos la matriz de adyacencia, para que finalmente aplicando funciones y comando orientados al algebra lineal en Matlab, simularemos el funcionamiento del motor de búsqueda Google.

Palabras claves: Crawlers, PAJEK, Cadenas de Markov, Matriz estacionaria, vector propio, método de las potencias.

RESUMO

No presente projeto de teses é apresentado o algoritmo de Pagerank de forma detalhado retomando a teoria de Markov, contemplando algumas definições e propriedades úteis para a compresão de algoritmo, bem como o método das potências para a obtenção de valores próprios e vectores próprios dominantes de uma matriz estocástica, assim como o teorema de Perron-Frobenius que sob as circunstâncias assegura a existência do vetor buscado (PageRank do vetor); Com o objetivo de fornecer um princípio teórico para o entendimento de algoritmo. Enseguida, para fim de entender as diferentes situações que o modelo pode enfrentar, uma simulação será apresentado neste trabalho. Em particular estamos interessados em construir de forma aproximada um buscador tipo Google. Em primeiro capítulo de esta teses enunciaremos o cimento matemático necessário para o entendimento do algoritmo, vamos a explicar Cadeias de Markov, Norma de vetores, a dependência ea independência linear, os valores e os vetores próprios, a multiplicidade algebraica e geometrica, a matriz diagonalizável, o teorema de Perron-Frobenius e método de potências; Seguidamente detalaremos definições de Web Crawler assim como o funcionamento do mesmo, fazendo uso do software OpenWebSpider, depois abordaremos o software PAJEK, o que nos permite visualizar nosso espaço web analizado por o Crawler e finalmente daremos um breve resumo do uso do Matlab, em particular orientado al algebra linear. No segundo capítulo detalharemos o algoritmo de ordenação usado por Google e aplicá-lo matematicamente, fazendo um planejamento do modelo. No terceiro capítulo, por meio de casos de estudos e de análises associadas a uma rede que consta de cinco e quatro paginas, detalharemos o algoritmo de ordenamento do Google, usando as definições vistas nos capítulos preliminares. A última parte será dedicada a uma simulação do algoritmo, detalharemos as ações que realizaram o Crawler, em seguida através da integração entre PAJEK e Matlab generemos a matriz de adjacência finalmente aplicando funções e comandos orientados a álgebra linear em Matlab, simularemos o funcionamento do motor de busca Google.

Palabras claves: Crawlers, PAJEK, Cadeias de Markov, Matriz estacionaria, vetor próprio, método das potências.

ABSTRACT

In the present dissertation project, the PageRank algorithm is presented in detail, retaking the Markov chain theory, considering some definitions and useful properties for the compression of the algorithm, as well as the method of the powers to obtain the eigenvalues and Dominant vectors of a stochastic matrix, as well as the Perron-Frobenius theorem that under certain conditions assures us the existence of the searched vector (PageRank vector); With the aim of providing a theoretical principle for the understanding of the algorithm. Then, in order to understand the different situations that the model may face, a simulation will be presented in this paper. In particular, we are interested in constructing an approximate Google-type search engine. In the first chapter of this thesis we will enunciate the mathematical foundation necessary for the understanding of the algorithm, we will explain Markov Chain, Vector norm, Dependence and linear independence, eigenvalues and eigenvectors, algebraic and geometric multiplicity, diagonizable matrix, Perron-Frobenius theorem And methods of the powers; Then we will detail Web Crawler definitions as well as its operation, making use of the OpenWebSpider software, then we will approach the software PAJEK, which allows us to visualize our web space analyzed by the Crawler and finally we will give a brief detail of the use of Matlab, in Particularly oriented to linear algebra. In the second chapter we will detail the sort algorithm used by Google and apply it mathematically, making a model approach. In the third chapter, through two case studies and two graphs associated with a network consisting of 5 and 4 pages, we will detail the Google ranking algorithm, using the definitions seen in the preliminary chapters. The last part will be dedicated to the simulation of the algorithm, for this we will detail the actions that will perform the Crawler, then through the integration between PAJEK and Matlab we generate the adjacency matrix, so finally applying functions and commands oriented to linear algebra in Matlab, We will simulate the operation of the Google search engine.

Key words: Crawlers, PAJEK, Markov chains, Stationary matrix, own vector, power method.

JURADO CALIFICADOR

LIC MG. MANUEL TUESTA MORENO

PRESIDENTE

ING. ALEJANDRO REATEGUI PEZO

MIEMBRO

ING. JOSE EDGAR GARCIA DIAZ

MIEMBRO

ING.CARLOS ALBERTO GARCIA CORTEGANO

ASESOR

UNIVERSIDAD NACIONAL DE LA AMAZONIA PERUANA
FACULTAD DE INGENIERIA DE SISTEMAS E INFORMATICA

ACTA DE SUSTENTACION DE TESIS

Siendo las 19:00 horas del día 11 de Marzo del 2017, en la Instalación del Auditorio de la Facultad de Ingeniería de Sistemas de la Universidad Nacional de la Amazonia Peruana, sito en calle Moore N° 280 – Iquitos, se instaló el jurado examinador integrado por los siguientes miembros:

Presidente : Lic. Manuel Tuesta Moreno
Primer Miembro : Ing. José Edgar García Díaz
Segundo Miembro : Ing. Alejandro Reátegui Pezo

SECRETARIA ACADEMICA FISI - UNAP

Acto seguido, se recibió al Tesista: **JULIO MARTIN ROJAS TENAZOA,** quien sustento el Tema: **"Aproximación Matemática y Computacional del Motor de Búsqueda Google"**, para optar el Título Profesional de Ingeniero de Sistema e Informática, de acuerdo a lo establecido en el Reglamento de Grados y Títulos de la Facultad de Ingeniería de Sistemas e Informática.

Posteriormente, al Acto de sustentación de la Tesis, se procedió al cálculo de Calificación obteniéndose el siguiente resultado:

	Calificaciones	
	En número	**En letras**
La Ejecución de la Tesis	16.00	Dieciseis y 00/100
La Sustentación de la Tesis.	15.87	Quince y 87/100
Promedio Final	15.94	Quince y 94/100

De la evaluación se desprende que el resultado es:

Aprobado

Con las siguientes observaciones:

Las que se indican en el informe final.

Se da por concluido el acto de sustentación a las 20:30 horas, procediéndose a firmar el Acta .

Lic. Manuel Tuesta Moreno
Presidente

Ing. José Edgar García Díaz
Primer Miembro

Ing. Alejandro Reátegui Pezo
Segundo Miembro

ÍNDICE GENERAL

I. INTRODUCCION XVI
II. ANTECEDENTES XIX
III. OBJETIVOS XXI
A. OBJETIVO GENERAL XXI
B. OBJETIVOS ESPECIFICOS XXI
IV. JUSTIFICACION XXII
CAPÍTULO I 24
MARCO TEÓRICO 24
1.1. FUNDAMENTOS MATEMATICOS 24
1.1.1. CADENA DE MARKOV 24
1.1.1.1. MATRIZ ESTACIONARIA 27
1.1.1.2. MATRIZ ESTACIONARIA EN VARIOS PASOS 28
1.1.1.3. CADENAS IRREDUCTIBLES 29
1.1.2. NORMA DE VECTORES 30
1.1.2.1. NORMALIZACION DE UN VECTOR 30
1.1.2.2. NORMA DE FROBENIUS 31
1.1.3. DEPENDENCIA E INDEPENDENCIA LINEAL 32
1.1.4. VALOR PROPIO Y VECTOR PROPIO 33
1.1.4.1. CALCULO DE VALORES PROPIOS 33
1.1.4.2. MULTIPLICIDAD ALGEBRAICA Y GEOMETRICA 34
1.1.4.3. CALCULO DE VECTORES PROPIOS 36
1.1.4.4. PROPIEDADES DE LOS VALORES PROPIOS [16] y [17] 37
1.1.4.5. PROPIEDADES DE LOS VECTORES PROPIOS [18] 37
1.1.5. MATRIZ DIAGONIZABLE 38
1.5.1.1. PROPIEDADES DE UNA MATRIZ DIAGONIZABLE [10] y [15] 40
1.1.6. TEOREMA DE PERRON-FROBENIUS 41
1.1.7. METODO DE LAS POTENCIAS 44
1.2. WEB CRAWLER 47

1.2.1. DEFINICION 47
1.2.2. BREVE HISTORIA DE LOS WEB CRAWLER 47
1.2.3. FUNCIONAMIENTO DE LOS WEBS CRAWLER 49
1.2.4. USO DE LOS WEB CRAWLERS.....50
1.3. INTRODUCCION AL ENTORNO DE OPENWEBSPIDER(JS)V.0.2.3.....52
1.3.1. REQUERIMIENTOS 52
1.3.2. CONFIGURACION 52
1.3.3. CONFIGURACION DE LA BASE DE DATOS 54
1.3.4. RASTEAR URLS.....58
1.4. INTRODUCCION AL ENTORNO DE PAJEK 63
1.4.1. DEFINICIÓN 63
1.4.2. INICIANDO PAJEK.....64
1.4.3. GENERANDO GRAFOS EN PAJEK 65
1.4.4. CENTRALIDAD.....68
1.5. INTRODUCCION AL MATLAB.....71
1.5.1. DEFICION 71
1.5.2. ENTORNO.....72
1.5.3. OPERADORES BASICOS.....74
1.5.4. DECLARACIÓN DE VARIABLES.....75
1.5.5. MATRICES Y VECTORES 76
1.5.6. OPERACIONES ESPECIFICAS CON MATRICES 78
1.5.7. VALOR Y VECTOR PROPIO 79
1.5.8. PROGRAMACION EN MATLAB 80
1.6. BUSCADORES.....85
1.6.1. DEFINICION 85
1.7. GOOGLE.....87
1.7.1. BREVE HISTORIA DE GOOGLE [38]. 87
1.7.2. MEDICION DE LA IMPORTANCIA DE UN DOCUMENTO WEB [36] y [39]. 90
CAPÍTULO II.....92
EL ALGOTIRMO DE ORDENACION DE GOOGLE.....92
2.1. PLANTEAMIENTO DEL MODELO.....93

2.2. EL ALGORITMO PAGERANK Y EL TEOREMA DE PERRON-FROBENIUS ...97
2.3. CALCULO DEL VECTOR PAGERANK 106
CAPÍTULO III 108
PLANTEAMIENTO DE LOS CASOS DE ESTUDIO 108
3.1. CASO I: CALCULO DEL VECTOR DE IMPORTANCIAS 108
3.2. CASO II: CALCULO DEL PAGERANK CON EL METODO DE LAS POTENCIAS 115
CAPÍTULO IV 124
SIMULACIÓN 124
4.1. INICIAR EL RASTREO DE LA PAGINAS 124
4.2. CONSTRUIR EL ARCHIVO .NET DE PAJEK....125
4.3. CONSTRUIR EL GRAFO DE CONEXIONES EN PAJEK 128
4.4. CONSTRUCCIÓN DE LA MATRIZ DE ADYACENCIA....136
4.5. ASIGNACION DEL PAGERANK DE CADA NODO(VERTICE) 137
CAPÍTULO V 144
CONCLUSIONES Y RECOMENDACIONES 144
5.1. CONCLUSIONES 144
5.2. RECOMENDACIONES....145
BIBLIOGRAFÍA 154

ÍNDICE DE FIGURAS

Figura 1. 1. Trayectoria de un espacio estocástico 25
Figura 1. 2. Oskar Perron y Ferdinand Georg Frobenius 41
Figura 1. 3. Arquitectura del primer Web Crawler 48
Figura 1. 4. Funcionamiento de un Web Crawler 50
Figura 1. 5. Ventana inicial de la aplicación 53
Figura 1. 6. OpenWebSpider(js) v0.23 53
Figura 1. 7. Opciones para la configuración de la base de datos. 55
Figura 1. 8. Opciones para la configuración de la base de datos-II. 56
Figura 1. 9. Creación de la base de datos y sus respectivas tablas. 57
Figura 1. 10. Vista de las bases de datos y sus respectivas las tablas. 57
Figura 1. 11. Ventana principal para rastrear Urls. 58
Figura 1. 12. Ventana para rastrear la Url www.unapiquitos.edu.pe. 60
Figura 1. 13. Proceso de rastreo de la url asignada www.unapiquitos.edu.pe. 61
Figura 1. 14. Vista de la Tabla *pages* de la base de datos *ows_index*. 62
Figura 1. 15 Andrej Mrvar y Vladimir Batagelj 63
Figura 1. 16. Ventana principal de Pajek 64
Figura 1. 17. Creación del archivo txt para declarar los vértices y los edges. 65
Figura 1. 18 Abrir archivos .NET en Pajek. 66
Figura 1. 19. Reporte del archivo .NET seleccionado. 67
Figura 1. 20. Acciones para Visualizar el grafo generado en Pajek. 67
Figura 1. 21.Grafo del archivo "*Ejemplo Tesis JMRT.net*" en Pajek. 68
Figura 1. 22. Acciones para generar un grafo con centralidad 69
Figura 1. 23. Grafo con centralidad del archivo "Ejemplo Tesis JMRT.net" 69
Figura 1. 24. Ventana principal de Matlab R2015a 72
Figura 1. 25. Línea de comandos del sistema. 72
Figura 1. 26 Ventana Command history 73
Figura 1. 27 Ventana Workspace 73
Figura 1. 28.Ventana Current Folder. 74

Figura 1. 29. Programa para resolver una ecuación de segundo grado. 81
Figura 1. 30. Programa para resolver una ecuación de segundo grado. 82
Figura 1. 31. Programa con función para resolver una ecuación de segundo grado. 84
Figura 1. 32 . Esquema básico de la arquitectura de un buscador.....86
Figura 1. 33. Lawrence Page y Sergei Brin.....87
Figura 1. 34. Página principal de Google en septiembre de 1998 89
Figura 1. 35. Ejemplo grafico del PageRank.....91

Figura 2. 1.Web de cinco páginas con sus respectivos enlaces.....94
Figura 2. 2. El grafo de conexiones correspondientes.....95
Figura 2. 3.Web de cuatro páginas con sus respectivos enlaces. Caso 1.....100
Figura 2. 4.Web de cuatro páginas con sus respectivos enlaces. Caso 2.....101
Figura 2. 5. Web de cuatro páginas con sus respectivos enlaces. Caso 3.....103

Figura 3. 1.Web de cinco páginas con sus respectivos enlaces. Caso práctico II 115

Figura 4. 1. Ventana de iniciación del rastreo 124
Figura 4. 2. Entorno grafico Mysql Workbench con las tablas ows_host y ows_index.....125
Figura 4. 3. Consulta SQL para generar el archivo .net 126
Figura 4. 4. Mensaje indicando que la migración fue satisfactoria 126
Figura 4. 5. Lista de vértices del archivo .net.....127
Figura 4. 6. Lista de los Arcs del archivo .net.....127
Figura 4. 7. Reporte del archivo Tesis.net.....128
Figura 4. 8. Opciones **Layout Enegery/Fruchterman Reingold/3D**.....129
Figura 4. 9. Grafo del archivo Tesis.net en 3D 129
Figura 4. 10. Opciones **Options/ Mark Vertices Using /No Labels**.....130
Figura 4. 11.Grafo del archivo Tesis.net sin etiquetas ni números 130
Figura 4. 12. Opciones **Network/Create New Network/ Transform/Reduction/Degree/All** 131

Figura 4. 13. Ventana Minimum degree of vertices 131
Figura 4. 14. Reporte del archivo Tesis.net simplificado 132
Figura 4. 15. Opciones Options/ Mark Vertices Using /Numbers 132
Figura 4. 16. Grafo del archivo Tesis.net simplificado 133
Figura 4. 17. Ventana para guardar el grafo simplificado 133
Figura 4. 18. Reporte del archivo TesisFinal.net 134
Figura 4. 19. Lista de los vértices del archivo TesisFinal,net 134
Figura 4. 20. Lista de los Arcs del archivo TesisFinal.net 134
Figura 4. 21. Opciones **Network/Create Vector/Centrality/Weighted Degree/All** 135
Figura 4. 22. Grafo del archivo TesisFinal.net con centralidad 135

ÍNDICE DE TABLAS

Tabla 1. 1. Argumentos y acciones para configurar la base de datos en OpenWebSpider(js). .54
Tabla 1. 2 . Argumentos y acciones para rastrear urls en OpenWebSpider(js). 59
Tabla 1. 3. Vector fila y vector columna en MATLAB 76
Tabla 1. 4. Operación de matrices con operadores aritméticos en MATLAB 77
Tabla 1. 5 Operaciones específicas de matrices en MATLAB.....78
Tabla 1. 6. Operaciones específicas de matrices en MATLAB.....79

Tabla 4. 1. Lista de las 50 páginas con mejor PageRank 141

I. INTRODUCCION

Encontrar una página, que atienda los intereses del internauta por algún asunto específico es algo muy valioso en el internet. Hasta 1997/98, la mejor tecnología para ese fin era ofrecida por el buscador AltaVista. Esa tecnología se basaba en tres principios que pueden ser sintetizados por:

1. Uso de programas (Crawler/Spider) para recorrer las páginas web, registrando todas las palabras y los enlaces existentes de forma recursiva.
2. Registro y organización de las páginas recuperadas por el Crawler, extrae una representación interna de la misma y la vuelca en forma de índice en una base de datos.
3. Realización de un ranking de la importancia o popularidad relativa de cada página, de este modo cuando son hechas las búsquedas, las páginas que se encuentran en las listas representan contenido más útil y satisfactorio para el usuario.

AltaVista avanzo bastante en los primeros principios, pero la solución propuesta para el tercer principio no era la más adecuada. En poco tiempo diversos mecanismos fueron creados para adulterar el proceso de elaboración del ranking, disminuyendo con eso el valor de las sugestiones de las páginas ofrecidas por AltaVista.

La solución más definitiva para el tercer principio fue el mecanismo de búsqueda visto en 1997/98 con la propuesta del algoritmo del PageRank.

Este algoritmo fue creado por el matemático Sergey Brin y un informático Larry Page cuando estudiaban su doctorado en la Universidad de Stanford, ambos posteriormente fundaron la compañía Google. Gran parte del éxito se debe a la calidad de su mecanismo para establecer el ranking de importancia de las páginas sugeridas para los usuarios, fundamentado en el algoritmo de PagerRank.

El algoritmo utiliza la basta estructura de enlaces como un indicador del valor de una página individual. PageRank interpreta el enlace de una página A a una página B como un voto. Pero el PageRank va más allá de contar el número de votos o enlaces que recibe una página; también analiza la página que emite el voto. Los votos emitidos por páginas que son importantes pesan más y ayudan a volver a otras páginas importantes.

Este algoritmo se fundamenta en elementos asociadas a las cadenas de Markov, siendo la solución obtenida a partir del vector propio dominante de una matriz estocástica específica, utilizando el método de las potencias.

En esta tesis se abordara al algoritmo de PageRank de forma detallada retomando la teoría de cadena de Markov, contemplando algunas definiciones y propiedades útiles para la compresión del algoritmo, bien como el método de las potencias para la obtención de los valores propios y vectores dominantes de una matriz estocástica, así como el teorema de Perron-Frobenius, que bajo ciertas condiciones nos asegura la existencia del vector buscado(vector PageRank) ; con el objetivo de proveer un principio teórico para el entendimiento del algoritmo.

Enseguida, a fin de entender las diferentes situaciones que el modelo puede enfrentar, una simulación será presentada en este trabajo. En particular estamos interesados en construir de manera aproximada un buscador tipo Google para ello se procederá a:

- Recolectar información del Web mediante un Crawler, es decir utilizaremos el software OpenWebSpider. Este Crawler recorrerá la dirección web indicada capturando e insertando en una base de datos los enlaces y las relaciones existentes entre ellas, de esta manera dispondremos toda la información de las relaciones entre las páginas web.

- Enseguida, se creará un grafo de conexiones; para este fin utilizaremos el software PAJEK. Este diseñador de grafos nos permite visualizar nuestro espacio web analizado por el Crawler.

- Concluido este proceso, mediante las relaciones entre páginas webs, generamos la matriz de adyacencia (matriz cuadrada cuyos elementos son 0 y 1), que nos permitirá aplicar el algoritmo de ordenamiento de PageRank de Google, mediante los principios matemáticos mencionados.

- Y finalmente simularemos el funcionamiento del motor de búsqueda Google, de este modo cuando es hecha la lista, las páginas que se encuentran ordenadas representan contenido más útil y satisfactorio para el usuario.

II. ANTECEDENTES

Debido a la prominencia de Google como mecanismo de búsqueda, su sistema de ordenación, tuvo una profunda influencia en el desarrollo y la estructuración de internet, y que tipos de información y servicios son visitados con más frecuentemente.

Pereira (2009), del Instituto de Matemática y Estadística de la Universidad de São Paulo en su tesis: "Ordenación de las Páginas de Google- "Page Rank"", en esta investigación es presentado una revisión de la cadena de Markov contemplando algunas definiciones y propiedades útiles para la comprensión del algoritmo, bien como el método de las potencias para la obtención de los autovalores y autovectores de una matriz, con el objetivo de proveer cimentación teórico para el entendimiento del PageRank.

Pais (2009), del programa de Pos-Graduación e n ciencias de la computación de la Universidad Federal de Minas Gerias en su tesis: "Exploración de la popularidad para búsquedas de información en blog", las principales contribuciones de esta tesis son:

- Investigar la capacidad de los motores de búsqueda actuales en explorar la popularidad de los blog para evidenciar posibles mejoras que pueden ser hechas debido a las características particulares de los blog.

- Presentar una nueva forma de ordenar el resultado de las consultas hechas a un motor de búsqueda para blogs utilizando el PageRank como un factor más en el cálculo de la similitud entre la consulta y el documento.

Motta (2007), del Instituto de ciencias exactas de la Universidad Federal de Itajubá en su tesis "Como funcionan los Search Engines" en este trabajo se abordará las tres grandes áreas de la arquitectura del search engine (motores de búsqueda o buscadores) y presentará un ejemplo ilustrativo de esta arquitectura. Las tres áreas de un motor de búsqueda consisten en:

- Web Crawler, donde las informaciones son capturadas.

- Indexación responsable por la organización y clasificación de las informaciones.

- La búsqueda, donde el usuario requisa informaciones y el search engine retorna un conjunto de datos referentes a (a los) termino(s) informados.

- Ofrecer, a raíz de esos resultados, recomendaciones prácticas para que una página web logre posiciones destacadas en los resultados de búsquedas de Google.

Adrian (2015), de la Universidad de la Rioja, Facultad de Ciencias Estudios Agroalimentarios e Informática, departamento de matemáticas y computación, en su tesis "El teorema de Perron-Frobenius y su aplicación en el algoritmo de búsqueda de Google", el objetivo de este trabajo es mostrar como resultados matemáticos profundamente abstractos y teóricos pueden llegar a tener aplicaciones prácticas realmente sorprendentes. En concreto veremos como el algoritmo de ordenamiento de Google, sin duda una de los buscadores más potentes y utilizados que existen, basa su funcionamiento en el teorema de Perron.Frobenius, un resultado de algebra lineal relacionados con matrices irreductibles.

Torres (2016), de la Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias Físico Matemáticas, en su tesis "Cadenas de Markov aplicadas al ordenamiento de páginas web" en este trabajo se presenta una pequeña introducción de las cadenas de Markov, que luego ayudara a la construcción de la matriz Google, además de la obtención del vector PageRank , enseguida se mostrara que la matriz Google como es de transición entonces, convergerá a un vector propio dominante, que es el PageRank.

III. OBJETIVOS

A. OBJETIVO GENERAL

- Aproximar matemática y computacionalmente al funcionamiento del motor de búsqueda Google.

B. OBJETIVOS ESPECIFICOS

- Proveer un principio teórico matemático para el entendimiento del algoritmo PageRank.

- Adaptar un Crawler, que recorrerá las diferentes páginas de forma automática, para luego almacenar en una base de datos, toda la información de las páginas y sus enlaces existentes.

- Diseñar un grafo de conexiones, que nos permitirá visualizar las relaciones que existe entre las páginas según el espacio analizado por el Crawler.

- Diseñar la matriz adyacente, según las relaciones que existe entre las páginas.

- Mediante los principios matemáticos y técnicas de programación, se diseñará y simulará el funcionamiento del motor de búsqueda Google, en particular utilizaremos Matlab para tal fin.

IV. JUSTIFICACIÓN

Las búsquedas en los principales motores, son hoy uno de los fenómenos más importantes de Internet. Tienen importancia social, económica y política. Forman parte de la vida cotidiana de la gran mayoría de los usuarios. Entre los diferentes buscadores, Google ocupa una posición de predominio mundial.

No hemos hallado estudios académicos en la amazonia peruana y probablemente en el Perú sobre cómo funciona el buscador Google. La investigación preliminar nos indica que nadie ha abordado con rigor este tema. En la página Red Peruana de Tesis Digitales, tan solo aparece una tesis de maestría que contenga Google en su comprendido. Se trata de la tesis de Héctor Manuel Bedón Monzón "*Prototipo de búsqueda en Internet basado en algoritmos genéticos*" que presenta el diseño y la implementación de un prototipo de agente inteligente para la extracción de información sobre una plataforma Linux basándose en métodos de algoritmos genéticos. Por lo tanto, no tiene ninguna relación con el funcionamiento del buscador. En la página Cybertesis, que contiene información sobre tesis de unas 16 universidades del Perú, no ofrece resultados que contenga la palabra Google o PageRank. Además, haciendo una búsqueda en la web sobre el tema, aparecen algunos artículos y páginas web del Perú, siendo la más relevante el artículo titulado "PageRank, el corazón de Google" elaborado por un miembro de la Sociedad de Estudiantes de Ciencias de la Computación en el Perú (SECC), que no aporta mucha información al respecto; lo único que hemos encontrado que aborde el tema son libros, artículos y tesis universitarias (en particular de países como EEUU, Brasil y España).

Dada la insuficiente información que existe en nuestro medio con respecto al funcionamiento del buscador Google, se consideró pertinente realizar esta tesis con el fin de dar a conocer por medio de una aproximación matemática y computacional de como Google ordena los resultados encontrados al momento de realizar una búsqueda.

V. APORTES

Un aporte que puede llegar a ofrecer esta tesis, es el de proveer conocimiento a quien lo consulte respecto de cómo implementar un buscador de tipo Google y adaptarlo a las necesidades que se le presenten según el problema al que quiera darle solución.

Un beneficio extra consiste en poder adaptar este buscador a una empresa que tenga la intención de proporcionar a sus clientes una herramienta de búsqueda dentro de su organización.

CAPÍTULO I

MARCO TEÓRICO

1.1. FUNDAMENTOS MATEMATICOS

1.1.1. CADENA DE MARKOV

En esta sección se exponen algunos conceptos esenciales sobre cadenas de Markov[1] considerados necesarios para el desarrollo de la tesis. Se comenzará definiendo algunos conceptos esenciales para luego tratar brevemente las Cadenas de Markov propiamente dichas. Las definiciones y las propiedades aquí citados se encuentran en [1], [2] y [3]

Definición. Es un proceso estocástico a tiempo discreto $\{x_n : n = 0,1,2 \dots\}$ con espacios de estados discretos $S = \{0,1, \dots\}$, que sin pérdida de generalidad supondremos que consta de los primeros enteros a partir del 0 y que satisface la propiedad de Markov.

$$P(x_{n+1} = x_{n+1} \mid x_0 = x_0 \dots x_n = x_n) = P(x_{n+1} = x_{n+1} \mid x_n = x_n) \qquad \textbf{(1.1)}$$

Esto es para cualquiera estado x_0, x_1 hasta x_{n+1} y para cualquier valor entero de n a partir de 0, la probabilidad que x_{n+1} tome el valor de x_{n+1} , dado que el proceso ha pasado por los estados x_0, x_1 hasta x_n, es simplemente la probabilidad del mismo evento dado $x_n = x_n$.

La propiedad de Markov significa que dado que se conoce la trayectoria del proceso hasta el tiempo n, la distribución de la variable x_{n+1} depende únicamente del ultimo valor observado, esto es x_n y no de los valores anteriores (figura 1)

[1] Andrei Andreyevich Markov fue un matemático ruso nacido en Ryazan, el 14 de junio de 1856 y fallecido en Petrogrado (ahora San Petersburgo), Rusia, el 20 de julio de 1922

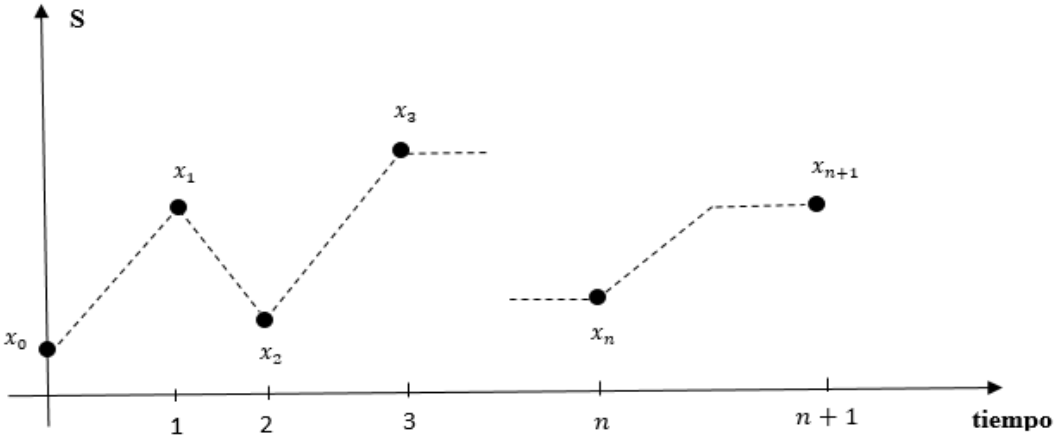


Figura 1. 1. Trayectoria de un espacio estocástico

Supongamos que el proceso comienza en x_0 mediante esta ley de probabilidad, cuando n es igual a 0 se genera al azar el estado del proceso al tiempo 1, a partir de x_1 se genera el nuevo estado x_2 , a partir de x_2 se genera un nuevo estado x_3 y asi sucesivamente. Los valores subsecuentes se generan al azar de acuerdo a esta ley de probabilidad condicional.

El aspecto importante de este proceso es entonces, esta condición llamada propiedad de Markov la cual condicionando de manera sucesiva es equivalente a la siguiente condición:

$$\boldsymbol{P}(x_0 = x_0, x_1 = x_1 \dots x_{n+1} = x_{n+1}) = P(x_0 = x_0).P(x_1 = x_1|x_0 = x_0) \dots P(x_{n+1} = x_{n+1}|x_n = x_n) \quad \textbf{(1.2)}$$

Esto significa que las probabilidades conjuntas de las primeras variables de una cadena de Markov, se pueden escribir como el producto de una distribución de probabilidad inicial, así llamaremos a la distribución de la variable x_0 y las probabilidades de pasar de un estado a otro en dos tiempos sucesivos. Estos elementos son importantes; escribiremos su definición:

$$\{P(x_0 = 0), P(x_0 = 1), P(x_0 = 2), \dots\} \quad \textbf{(1.3)}$$

A la distribución de probabilidad de la variable aleatoria x_0, se le llama distribución inicial de la cadena de Markov.

Entonces asociado al estado 0, tenemos la probabilidad de que x_0 tome el valor 0; para el estado 1 tenemos la probabilidad de que x_0 tome el valor de 1; después la probabilidad de que x_0 tome el valor de 2 y así sucesivamente, entonces a este conjunto de probabilidades se le llama distribución inicial de la cadena de Markov, en general supondremos que esta distribución es conocida, en algunas situaciones supondremos que la cadena inicia en un valor determinado y esto significa que la distribución inicial está concentrada en un solo punto.

Entonces a partir de un valor inicial de la cadena de Markov se generan al azar los subsecuentes valores de acuerdo a las siguientes probabilidades de transición:
A la probabilidad:

$$P(x_{n+1} = j \mid x_n = i) = P_{ij}(n, n+1) \qquad \textbf{(1.4)}$$

Se le llama probabilidad de transición del estado i en el tiempo n, al estado j en el tiempo $n+1$. A estas probabilidades se les conoce como probabilidades de transición en un paso.

En general estas probabilidades se mantienen constantes para cualquier valor de n , en este caso se dice que la cadena es estacionaria u homogénea en el tiempo.

Las probabilidades de transición $P_{ij}(n, n+1)$ son estacionarias en el tiempo si no depende de n, es decir son idénticas a $P_{ij}(0,1)$ o bien $P_{ij}(1,2)$ o de $P_{ij}(2,3)$, etc.

En este caso omitiremos el argumento $(n, n+1)$ y le escribiremos como $P_{ij}(1)$, y cuando se trata de paso en general se omite, al menos que se quiera hacer énfasis en eso.
Entonces eso se escribe como:

$$P_{ij}(n, n+1) = P_{ij} \qquad \textbf{(1.5)}$$

Considerando los distintos valores de i y j en el espacio de estados, se obtiene la matriz de probabilidades de transición en un paso.

1.1.1.1.MATRIZ ESTACIONARIA

A continuación, solo consideraremos cadenas con probabilidades de transición (estacionarias), está dada como la siguiente matriz cuadrada.

$$P = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \vdots \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & \dots \end{array} \\ \begin{pmatrix} P_{00} & P_{01} & P_{02} & \dots \\ P_{10} & P_{11} & P_{12} & \dots \\ P_{20} & P_{21} & P_{22} & \dots \\ \vdots & \vdots & \vdots & \end{pmatrix} \end{array}$$

Donde:

$$P_{ij} = P(x_{n+1} = j \mid x_n = i) \qquad \textbf{(1.6)}$$

La entrada i, j de esta matriz, es la probabilidad P_{ij}(fila i, columna j) y corresponde a la probabilidad de que el proceso pase del estado i al estado j en un tiempo o un paso; es decir el estado i fila es el estado de partida y el estado j columna es el estado destino. Esta matriz presenta las siguientes 2 propiedades:

$$P_{ij} \geq 0 \qquad \textbf{(1.7)}$$

$$\sum_j P_{ij} = 1 \qquad \textbf{(1.8)}$$

Primero tenemos que las entradas de esta matriz son no negativas, esto es evidente puesto que se trata de probabilidades.

La segunda propiedad nos dice que la suma de las entradas de la matriz en una misma columna es 1, en otras palabras, esta propiedad establece que a partir de cualquier estado inicial i con probabilidad 1, la cadena pasa necesariamente algún elemento del espacio de estados al siguiente momento, entonces tenemos la siguiente definición:

A toda matriz cuadrada que cumpla las propiedades 1.7 y 1.8 se le llama matriz estocástica, de esta manera atraves de la definición de la cadena de Markov, se determinan tres elementos importantes:

- Un espacio de estados discretos S.
- Una distribución de probabilidad (inicial) sobre el espacio de estados.
- Una matriz estocástica.

Recíprocamente si tenemos un espacio de estados discreto S, una distribución de probabilidad sobre S y una matriz estocástica, se puede definir una cadena de Markov.

1.1.1.2.MATRIZ ESTACIONARIA EN VARIOS PASOS

A la probabilidad $P(x_{m+n} = j \mid x_m = i) = P_{ij}(m, m+n)$, se le llama probabilidad de transición del estado i en el tiempo m al estado j en el tiempo $m+n$. Esto es, dado que la cadena en el tiempo m se encuentra en el estado i, esta es la probabilidad de que después de n unidades de tiempo, la cadena se encuentre en el estado j

Observación. Como una consecuencia de la hipótesis de que las probabilidades de transición en un paso son estacionarias, tenemos que estas probabilidades de transición en n pasos también son estacionarias, esto es:

$$P_{ij}(m, m+n) = P_{ij}(0, n) \tag{1.9}$$

La probabilidad de pasar del estado i a j en los tiempos (m) ha $(m+n)$, coincide con la probabilidad de pasar del estado i al estado j en los tiempos 0 a (n) o 1 a $(n+1)$ o 2 a $(n+2)$, etc. Estas probabilidades entonces no dependen de m, y se escribe como:

$$P_{ij}(m, m+n) = P_{ij}(n) \tag{1.10}$$

No importando el momento en el cual se den eso n pasos; considerando los distintos valores de i y j en el espacio de estados se obtiene la matriz de probabilidades de transición en n pasos. La matriz de probabilidad de transición en n pasos para una cadena de Markov estacionaria, está dada por este arreglo:

$$P(n) = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ \vdots \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & \dots \end{array} \\ \begin{pmatrix} P_{00}(n) & P_{01}(n) & P_{02}(n) & \dots \\ P_{10}(n) & P_{11}(n) & P_{12}(n) & \dots \\ P_{20}(n) & P_{21}(n) & P_{22}(n) & \dots \\ \vdots & \vdots & \vdots & \end{pmatrix} \end{array} \tag{1.11}$$

Nuevamente la entrada ij de esta matriz es la probabilidad $P_{ij}(n_{ij})$ y corresponde a la probabilidad de pasar del estado i al estado j en n pasos; puede comprobarse que esta matriz es estocástica es decir tiene entradas no negativas y la suma de los elementos de sus columnas es 1.

1.1.1.3.CADENAS IRREDUCTIBLES

Se dice que una cadena de Markov es irreductible si todos los estados se comunican entre sí, es decir x accede a y para todo $x, y \in C$, donde C es un conjunto cerrado[2]

[2] Un conjunto de estados C en una cadena de Markov se dice que es cerrado si cualquier estado dentro de C puede alcanzarse desde cualquier otro estado de C y ningún otro estado fuera de C puede ser alcanzado desde cualquier estado dentro de C., es decir $p_{xy} = 0,\ x \in C\ y \notin C$

1.1.2. NORMA DE VECTORES

Sea $A = (x_1, x_2, \dots, x_n)^t$ un vector en $\mathbb{R}^n$, se define la norma o longitud de este vector, la cual denotaremos como $\|A\|$ al número real dado por $\|A\| = \sqrt{{x_1}^2 + {x_2}^2 + \cdots + {x_n}^2}$ [4], además cumple con las siguientes propiedades para cualquier vector $A\ y\ B\ \in\ \mathbb{R}^n$ y $\alpha \in \mathbb{R}$ [5]:

1. $\|A\| \geq 0\ y\ \|A\| = 0$ si y solo si $A = 0$
2. $\|\alpha A\| = |\alpha|\|A\|$
3. $\|A + B\| \leq \|A\| + \|B\|$ (desigualdad triangular)

1.1.2.1.NORMALIZACION DE UN VECTOR

Sea $A = (x_1, x_2, \dots, x_n)^t$ un vector[3] no nulo y no unitario, normalizar dicho vector es construir otro vector $\vec{A}$ a partir de este, que tenga la misma dirección, sentido y cuya norma sea la unidad. El proceso es multiplicar el vector A por el inverso de su norma. Dicho de otra manera:

$$\vec{A} = \frac{A}{\|A\|} = \frac{1}{\|A\|}A = \left(\frac{x_1}{\sqrt{{x_1}^2 + {x_2}^2 + \cdots + {x_n}^2}}, \frac{x_2}{\sqrt{{x_1}^2 + {x_2}^2 + \cdots + {x_n}^2}}, \dots, \frac{x_n}{\sqrt{{x_1}^2 + {x_2}^2 + \cdots + {x_n}^2}}\right)^t \qquad \textbf{(1.12)}$$

$$\|\vec{A}\| = \sqrt{\left(\frac{x_1}{\sqrt{{x_1}^2 + {x_2}^2 + \cdots + {x_n}^2}}\right)^2 + \left(\frac{x_2}{\sqrt{{x_1}^2 + {x_2}^2 + \cdots + {x_n}^2}}\right)^2 + \cdots + \left(\frac{x_n}{\sqrt{{x_1}^2 + {x_2}^2 + \cdots + {x_n}^2}}\right)^2} = 1$$

Ejemplo. Sea $A = (1,2,3)$, para normalizar aplicamos la ecuación 1.12

$$\vec{A} = \left(\frac{1}{\sqrt{1^2 + 2^2 + 3^2}}, \frac{2}{\sqrt{1^2 + 2^2 + 3^2}}, \frac{3}{\sqrt{1^2 + 2^2 + 3^2}}\right)^t = \left(\frac{1}{\sqrt{14}}, \frac{2}{\sqrt{14}}, \frac{3}{\sqrt{14}}\right)^t$$

Ahora la norma del vector $\vec{A}$ es la unidad $\Rightarrow$ $\|\vec{A}\| = \sqrt{\left(\left(\frac{1}{\sqrt{14}}\right)^2 + \left(\frac{2}{\sqrt{14}}\right)^2 + \left(\frac{3}{\sqrt{14}}\right)^2\right)} = 1$

[3] Consideramos el vector en columna, pero para facilitar la redacción de esta tesis, se escriben en filas acompañadas de un exponente *t* que indica la transpuesta, esto es la transformación de filas en columnas.

1.1.2.2.NORMA DE FROBENIUS

Dada una matriz A, su norma de Frobenius (llamada también la norma de Hilbert-Schmidt) [5], se define como:

$$\|A\|_F = \sqrt{traza(A^t A)} = \sqrt{\sum_{i,j=1}^{n} |a_{ij}|^2} \qquad \textbf{(1.13)}$$

Siendo $A = ((a_{ij})_{i,j=1\ldots n})$ y A^t la matriz transpuesta, además la traza se define como la suma de los elementos de la diagonal principal de la matriz $A^t A$

Ejemplo. Consideramos la matriz cuadrada**:**

$$A = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Calculamos $A^t A$

$$A^t A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 3 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 2 & 1 \\ 1 & 0 & 1 & 2 \end{pmatrix}$$

Aplicamos la ecuación 1.13

$$\|A\|_F = \sqrt{traza(A^t A)} = \sqrt{\sum_{i,j=1}^{n} |a_{ij}|^2} = \sqrt{(|3|)^2 + (|1|)^2 + (|2|)^2 + (|2|)^2} = \sqrt{18}$$

1.1.3. DEPENDENCIA E INDEPENDENCIA LINEAL

Un conjunto de vectores $\{v_1, v_2, \dots, v_n\}$ en un espacio vectorial V es linealmente independiente si existen escalares $c_1, c_2, \dots c_n$, al menos uno de los cuales no es cero [7] , tales que

$$c_1v_1 + c_2v_2 + \cdots + c_nv_n = 0 \text{ donde el cero (0) simboliza al vector nulo}$$

Un conjunto de vectores que no es linealmente dependiente se dice que es linealmente independiente. Dicho de otra manera:

El conjunto de vectores $\{v_1, v_2, \dots, v_n\}$ es linealmente independiente en un espacio vectorial V si y solo si [8]

$$c_1v_1 + c_2v_2 + \cdots + c_nv_n = 0 \text{ implica que } c_1 = 0, c_2 = 0, \dots, c_n = 0$$

Ejemplo. Sea los vectores $u = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix}$, $v = \begin{pmatrix} 1 \\ 3 \\ 0 \end{pmatrix}$, $w = \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix}$; buscamos tres valores x, y y z, que satisface la ecuación:

$$xu + yv + zw = x\begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} + y\begin{pmatrix} 1 \\ 3 \\ 0 \end{pmatrix} + z\begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Lo que equivale al sistema de ecuaciones siguiente:

$$\begin{array}{rcrcl} 2x + & y & + & z & = & 0 \\ & 3y & + & 2z & = & 0 \\ & & & 4z & = & 0 \end{array} \Rightarrow \begin{array}{r} x = 0 \\ y = 0 \\ z = 0 \end{array}$$

Dado que la única solución es trivial ($x = y = z = 0$) entonces los tres vectores son linealmente independientes.

1.1.4. VALOR PROPIO Y VECTOR PROPIO

Sea A una matriz cuadrada de orden n, se dice que un escalar λ es un valor propio, autovalores o eigenvalores de A si existe un vector v no nulo, de modo que [9]:

$$Av = \lambda v \tag{1.14}$$

Si λ es valor propio de A, entonces todos los vectores v que verifican la ecuación 1.14 reciben el nombre de vectores propios, autovectores o eigenvectores de A asociados al valor propio λ. Asociados a un valor propio dado, hay infinitos vectores propios (todos aquéllos proporcionales a uno dado, no nulo). Es decir, se verifica [10]:

$$Av - \lambda v = (A - \lambda I)v = 0 \tag{1.15}$$

Donde I es la matriz identidad (elemento neutro en producto de matrices) de orden $n\ x\ n$.

1.1.4.1.CALCULO DE VALORES PROPIOS

Teorema de Cayley-Hamilton[4] [11] y [12]. Establece que cada matriz cuadrada A satisface su ecuacion caracteristica, Si $p(\lambda) = det(A - \lambda I)$ [5] es el polinomio caracteristico de A, entonces $p(A)$ es la matriz nula. Es decir, $p(A) = 0$ **(1.16)**

---

[4] ***Arthur Cayley*** (Richmond, Reino Unido, 16 de agosto de 1821 - Cambridge, 26 de enero de 1895) fue un matemático británico. Es uno de los fundadores de la escuela británica moderna de matemáticas puras.

William Rowan Hamilton (4 de agosto de 1805- 2 de septiembre de 1865) fue un matemático, físico, y astrónomo irlandés, que hizo importantes contribuciones al desarrollo de la óptica, la dinámica, y el álgebra.

[5] Otra manera de expresarlo es $\boldsymbol{p(\lambda) = |A - \lambda I|}$

$$p(\lambda) = |A - \lambda I| = \left| \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} \right| = 0$$

$$p(\lambda) = |A - \lambda I| = \begin{vmatrix} a_{11} - \lambda & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} - \lambda & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} - \lambda \end{vmatrix} = 0 \qquad \textbf{(1.17)}$$

Al desarrollar el $p(\lambda) = |A - \lambda I| = 0$ obtenemos un polinomio de grado n del que tenemos que obtener los valores de λ que lo hacen cero.

$$P(\lambda) = a_0\lambda^n + a_1\lambda^{n-1} + \cdots + a_{n-1}\lambda + a_n = 0 \qquad \textbf{(1.18)}$$

Eso nos asegura que el numero de valores propios de A, a lo sumo es el grado n del polinomio $p(\lambda)$, si el polinomio $p(\lambda)$ tiene raices multiples, el número de valores propios distintos es estrictamente menor que n. Una vez obtenidas estas n raíces, digamos $\lambda_1, \lambda_2, \ldots, \lambda_n$, podemos obtener, para cada una de las que no están repetidas, un vector propio [13].

1.1.4.2.MULTIPLICIDAD ALGEBRAICA Y GEOMETRICA

Sea λ_i un valor propio de A, entonces la multiplicidad algebraica de λ_i, que se denota por $m_a(\lambda_i)$, es el número de veces que se repite como raiz del polinomio caracteristico. Cuando la multiplicidad algebraica es 1, λ_i es llamado valor propio simple [14].

El número maximo de vectores propios linealmente independientes que tienen asociados un valor propio λ_i, es decir la dimension del espacio nulo $N(A - \lambda I)$, se llama multiplicidad geometrica y se representa por $m_g(\lambda_i)$ [15]. Dicho de otra manera:

$$m_g(\lambda_i) = dim\left(V_{\lambda_i}\right) = \dim\left(N(A - \lambda_i I)\right) = n - rango(A - \lambda_i I) \qquad \textbf{(1.19)}$$

Donde n es el tamano de la matriz e I es la matriz identidad, ambos de tamaño nxn

Si λ_i es un autovalor de A con multiplicidad algebraica r , entonces $1 \leq dim(V_{\lambda_i}) \leq r$, es decir:

$$1 \leq m_g(\lambda_i) \leq m_a(\lambda_i) \qquad \textbf{(1.20)}$$

Lo que quiere decir que la multiplicidad geometrica de λ_i, esta comprendida entre 1 y la multpliplicidad algebraica λ_i, ademas no excede a su multiplicidad algebraica [15].

Observacion. Los valores propios con multiplicidad algebraica 1, su $dim(V_{\lambda_i})$ esta comprendido entre 1 y 1, por lo tanto la $m_g(\lambda_i) = 1$ sin tener que calcular el rango de la matriz.

Ejemplo. Consideramos la matriz $A = \begin{pmatrix} 5 & 1 & 0 & 0 \\ 0 & 5 & 1 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$

Calculamos los valores propios y su multiplicidad algebraica, resolviendo $|A - \lambda I| = 0$

$$\left|\begin{pmatrix} \lambda-5 & 1 & 0 & 0 \\ 0 & \lambda-5 & 1 & 0 \\ 0 & 0 & \lambda-5 & 0 \\ 0 & 0 & 0 & \lambda+1 \end{pmatrix}\right| = 0 \Rightarrow (\lambda-5)^3(\lambda+1) \Rightarrow \begin{cases} \lambda_1 = 5 & \text{con } m_a(\lambda_1) = 3 \\ \lambda_2 = -1 & \text{con } m_a(\lambda_2) = 1 \end{cases}$$

Para cada valor propio λ_i con sus multiplicidades algebraicas $m_a(\lambda_i)$, calcularemos los vectores propios asociados a $m_g(\lambda_i)$, empezaremos por el valor propio $\lambda = 5$ con multiplicidad algebraica 3, resolviendo $m_g(\lambda_1) = n - rango(A - \lambda_1 I)$

$$m_g(\lambda_1) = 4 - rango\left(\begin{pmatrix} 5 & 1 & 0 & 0 \\ 0 & 5 & 1 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} - 5\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}\right)$$

$$m_g(\lambda_1) = 4 - rango\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -6 \end{pmatrix} \Rightarrow m_g(\lambda_1) = 4 - 3 \Rightarrow m_g(\lambda_1) = 1$$

Las multiplicidades algebraicas y geométricas del valor propio $\lambda_1 = 5$, coinciden con la relación.

$$1 \leq m_g(\lambda_1) \leq 3$$

El valor propio $\lambda = -1$, por tener multiciplidad algebraica 1, su multiplicidad geométrica es 1 sin tener que calcular el rango de la matriz, por la tanto la relación estará comprendida.

$$1 \leq m_g(\lambda_2) \leq 1$$

1.1.4.3.CALCULO DE VECTORES PROPIOS

Los vectores propios asociados a un valor propio λ se calculan resolviendo el sistema de ecuaciones homogeneos [12].

$$\begin{pmatrix} a_{11} - \lambda & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} - \lambda & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} - \lambda \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \qquad \textbf{(1.21)}$$

$$\begin{cases} (a_{11} - \lambda)x_1 + a_{12}x_2 + \cdots + a_{1n}x_n & = 0 \\ (a_{21})x_1 + (a_{22} - \lambda)x_2 + \cdots + a_{2n}x_n & = 0 \\ \vdots & \\ (a_{n1})x_1 + (a_{n2})x_2 + \cdots + (a_{nn} - \lambda)x_n & = 0 \end{cases} \qquad \textbf{(1.22)}$$

Al resolver el sistema de ecuaciones, obtenemos los valores de $x_1, x_2, \dots, x_n$. Es decir:

$$v = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \tag{1.23}$$

1.1.4.4.PROPIEDADES DE LOS VALORES PROPIOS [16] y [17]

- La matriz A y su transpuesta A^t tienen los mismos valores propios.
- Si A es una matriz triangular entonces sus valores propios son los elementos de la diagonal principal.
- Dado $\alpha \in \mathbb{R}$, los valores propios de la matriz αA son $\alpha\lambda_1, \dots, \alpha\lambda_n$.
- Dado $k \in \mathbb{N}$, los valores propios de la matriz A^k son ${\lambda_1}^k, \dots, {\lambda_n}^k$.
- Si A es una matriz no singular entonces todos los valores propios son no nulos y los valores propios de la matriz A^{-1} son $\frac{1}{\lambda_1}, \dots, \frac{1}{\lambda_n}$
- Si A es una matriz ortogonal con valores propios reales, entonces sus valores propios son -1 o 1.
- Si A es una matriz idempotente entonces sus valores propios son 0 ó 1.
- La suma de los valores propios de A (incluidas las repeticiones) es igual a la traza de A.

$$Traza(A) = \lambda_1 + \cdots + \lambda_n$$

1.1.4.5.PROPIEDADES DE LOS VECTORES PROPIOS [18]

- Vectores propios asociados a valores propios distintos son linealmente independientes.
- Todo vector propio esta asociado a un unico valor propio.

1.1.5. MATRIZ DIAGONIZABLE

Una matriz A cuadrada de orden nxn es diagonalizable si y solo si tiene n vectores propios linealmente independientes. En este caso, la matriz diagonal D a la que es semejante es aquella cuyos elementos diagonales son los valores propios correspondientes. Por último, la matriz de paso P es la matriz cuyas columnas son los vectores propios. En particular, si A tiene n valores propios distintos, entonces es diagonalizable [10]. Dicho de otra manera

Se tiene que $A = PDP^{-1}$, con D diagonal, si y solo las columnas de P son vectores propios linealmente independientes de A, y los elementos diagonales de D son los valores propios de A, de modo que:

$$A = P\begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix}P^{-1}, \qquad P = (v_1|v_2|\dots|v_n) \tag{1.24}$$

Observacion. Para hallar las matrices D y P, sera preciso hallar con anterioridad los valores propios y los vectores propios de la matriz en mencion.

Ejemplo. Diagonalicemos $A = \begin{pmatrix} 3 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 3 \end{pmatrix}$

Calculamos los valores propios y su multiplicidad algebraica, resolviendo $|A - \lambda I| = 0$

$$\left|\begin{pmatrix} 3 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 3 \end{pmatrix} - \lambda\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\right| = 0 \Rightarrow (\lambda - 3)(\lambda - 1)(\lambda - 4)$$

Como todos los valores propios de A tienen multiplicidad algebraica 1, entonces son valores propios simples

Calculamos los vectores propios de A, resolviendo los sistemas $(A - \lambda I)x = 0$ y dando la solución en forma vectorial paramétrica, se obtienen bases de los espacios propios.

Para $\lambda_1 = 3$

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & -1 & -1 \\ 1 & -1 & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \Rightarrow \begin{cases} & -x_2 & & = 0 \\ -x_1 & -x_2 & -x_3 & = 0 \\ & -x_2 & & = 0 \end{cases} \Rightarrow \begin{cases} x_1 = -t \\ x_2 = 0 \\ x_3 = t \end{cases} \Rightarrow v_1 = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$$

Para $\lambda_1 = 1$

$$\begin{pmatrix} 2 & -1 & 0 \\ -1 & 1 & -1 \\ 0 & -1 & 2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \Rightarrow \begin{cases} 2x_1 & & -x_2 & & & = 0 \\ -x_1 & + & x_2 & & -x_3 & = 0 \\ & & -x_2 & + & 2x_3 & = 0 \end{cases} \Rightarrow \begin{cases} x_1 = t \\ x_2 = 2t \\ x_3 = t \end{cases} \Rightarrow v_2 = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

Para $\lambda_1 = 4$

$$\begin{pmatrix} -1 & -1 & 0 \\ -1 & -2 & -1 \\ 0 & -1 & -1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \Rightarrow \begin{cases} -x_1 & -x_2 & & & = 0 \\ -x_1 & -2\,x_2 & & -x_3 & = 0 \\ & -x_2 & + & -x_3 & = 0 \end{cases} \Rightarrow \begin{cases} x_1 = -t \\ x_2 = t \\ x_3 = -t \end{cases} \Rightarrow v_3 = \begin{pmatrix} -1 \\ 1 \\ -1 \end{pmatrix}$$

Las columnas de P son los vectores propios:

$$P = (v_1|v_2|v_3) = \begin{pmatrix} -1 & 1 & -1 \\ 0 & 2 & 1 \\ 1 & 1 & -1 \end{pmatrix}$$

Se colocan en la diagonal de D los valores propios en el orden correspondiente a como están colocados los vectores propios en P.

$$D = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 4 \end{pmatrix}$$

Se puede comprobar la semejanza examinando que $A = PDP^{-1}$

$$A = \begin{pmatrix} -1 & 1 & -1 \\ 0 & 2 & 1 \\ 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} 3 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 4 \end{pmatrix} \begin{pmatrix} -1/2 & 0 & 1/2 \\ 1/6 & 1/3 & 1/6 \\ -1/3 & 1/3 & -1/3 \end{pmatrix} \Rightarrow A = \begin{pmatrix} 3 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 3 \end{pmatrix}$$

Como cumple que $A = PDP^{-1}$, entonces la matriz A es diagonizable.

1.5.1.1.PROPIEDADES DE UNA MATRIZ DIAGONIZABLE [10] y [15]

- Si una matriz cuadrada A, de orden n, tiene n valores propios distintos (con multiplicidad algebraica y geometrica 1), entonces la matriz A es diagonizable.
- Una matriz cuadrada A, de orden n, es diagonalizable si y sólo si todos sus valores propios son reales y, además, la multiplicidad geométrica de cada uno coincide con su multiplicidad algebraica.
- Si la suma de las dimensiones del espacio vectorial es igual a la dimensión total (tamaño de la matriz nxn). Dicho de otra manera:

$$A \text{ es diagonizable } \Leftrightarrow \dim(v_{\lambda_1}) + \cdots + \dim(v_{\lambda_r}) = n \qquad \textbf{(1.25)}$$

1.1.6. TEOREMA DE PERRON-FROBENIUS

El teorema de Perron-Frobenius se debe a dos grandes matemáticos alemanes como son Oskar Perron (a la izquierda en la figura 1.2) y Ferdinand Georg Frobenius (a la derecha en la figura 1.2) [19].

Figura 1. 2. Oskar Perron y Ferdinand Georg Frobenius
Fuente: [19]

Oskar Perron fue un matemático alemán nacido el 7 de mayo de 1880 en Frankenthal y falleció el 22 de febrero de 1975 en Múnich. Hizo numerosas contribuciones en campos tan diversos como el Análisis, las Ecuaciones Diferenciales, el Algebra, la Geómetra, la Teoría de Números, etc., en los que publico algunos textos que se convertirán en clásicos [20].

Ferdinand Georg Frobenius fue un matemático alemán, nacido en Charlottemburg (en la actualidad Berlín) el 26 de octubre de 1849 y falleció en Berlín el 3 de agosto de 1917, es uno de los matemáticos más excelsos del siglo XX con aportaciones fundamentales en el álgebra

lineal y la teoría de ecuaciones diferenciales. Es sobre todo conocido por sus aportaciones a la teoría de grupos, y sus trabajos sobre las matrices no negativas pertenecen a la última etapa de su vida [20].

Oskar Perron publicó en 1907 un teorema que aseguraba la existencia de un valor propio simple, donde su vector propio asociado es estrictamente positivo, y fácilmente reconocible. La única inconveniencia era que las entradas de la matriz no podían ser iguales a 0. Sin embargo, en 1912 Ferdinand Georg Frobenius generalizó el teorema de Perrón para matrices no negativas [21].

En aquel momento este resultado sobre matrices irreducibles parecía algo bastante teórico y abstracto y ninguno de estos dos celebres matemáticos podía imaginar que años más tarde este teorema se convertiría en la base de innumerables aplicaciones. Hoy en día el teorema de Perron-Frobenius ha dado como resultado una importante teoría matemática que lleva su nombre y que tiene aplicaciones en diversas ramas de la ciencia, desde aplicaciones económicas, hasta el análisis de las características demográficas de una población, pasando por el estudio de la distribución de poder en una red social. Pero se tuvo que esperar casi 90 años hasta encontrar su aplicación más famosa, el motor de búsqueda de Google [19].

Antes de dar el teorema de Perron-Frobenius vamos a presentar una definición necesaria para establecer el enunciado del mismo.

Definición 1 [19]. Una matriz A de orden nxn se dice irreductible en uno de los siguientes casos:

- A es la matriz cero de orden $1x1$
- Si $n \geq 2$ y existe una matriz de permutación P tal que

$$PAP^T = \begin{pmatrix} B & 0 \\ C & D \end{pmatrix} \tag{1.25}$$

Donde B y D son matrices cuadradas de orden menor que n, P^T denota la transpuesta de P y 0 es la matriz cero. Si no existe tal P entonces se dice que A es irreductible.

Una vez definida el concepto de una matriz irreductible, presentaremos el enunciado del teorema de Perron-Frobenius.

Teorema 1 [19]. Sea A una matriz cuadrada de orden nxn con entradas no negativas $A \geq 0$ Si A es irreductible entonces:

- Existe un valor propio simple $\lambda > 0$, tal que $Av = \lambda v$, donde el vector propio correspondiente es $v > 0$. Además:
    a. $\lambda \geq |\mu|$ para cualquier otro valor propio μ de A
    b. El valor propio λ tiene multiplicidad algebraica y geométrica 1.
    c. Cualquier vector propio $\omega \geq 0$ no nulo de A es múltiplo de v.

- Si hay k valores propios de modulo máximo, entonces son las soluciones de la ecuación $\boldsymbol{x^k - \lambda^k = 0}$

Lo que nos dice este teorema a grandes rasgos, es que una matriz no negativa e irreductible siempre tiene un valor propio positivo de modulo máximo, asociado a un vector propio también positivo y del que son múltiplos todos los demás vectores propios positivos de la matriz. Este vector propio tendrá un papel muy importante en el algoritmo de ordenación como veremos en el capítulo II.

1.1.7. METODO DE LAS POTENCIAS

El método de las potencias es un método iterativo utilizado para la obtención del valor propio de modulo máximo y su correspondiente vector propio. Las definiciones y las propiedades aquí citados se encuentran en [22].

Se atribuye una aproximación inicial arbitraria para el vector propio correspondiente al valor propio dominante que es sucesivamente mejorado hasta que la precisión requerida sea encontrada. La convergencia para el valor propio dominante es simultáneamente obtenida.

Proposición. Sea A una matriz cuadrada de orden n y la cual tiene n vectores propios linealmente independientes (es decir, A es diagonizable) y un valor propio estrictamente dominante, es decir, si sus valores propios son:

$$\lambda_1, \lambda_2, \dots \lambda_n \tag{1.26}$$

Entonces ocurre que:

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots |\lambda_n| \tag{1.27}$$

No hay pérdida de generalidad en suponerlo ordenados, de forma que el mayor modulo se corresponde con λ_1.

Notaremos por v_i al vector propio asociado a cada valor propio λ_i para $i = 1,2, \dots n$ entonces por la hipótesis de la que hemos partido tenemos que: $\{v_1, v_2, \dots v_n\}$ son linealmente independiente. Esto significa que cualquier vector v del espacio vectorial $\mathbb{R}^n$ se puede expresar como combinación lineal de los vectores $\{v_1, v_2, \dots v_n\}$.
Es decir, existirán valores $a_i \in \mathbb{R}$ tales que:

$$v = \sum_{i=1}^{n} a_i \, v_i \tag{1.28}$$

Por lo tanto, tenemos que:

$$Av = \sum_{i=1}^{n} a_i \, Av_i = \sum_{i=1}^{n} a_i \, \lambda_i v_i \qquad \textbf{(1.29)}$$

Es decir, tenemos:

$$Av = a_1\lambda_1 v_1 + a_2\lambda_2 v_2 + \cdots a_n\lambda_n v_n \qquad \textbf{(1.30)}$$

Si multiplicamos esta expresión de nuevo por A, tenemos:

$$A^2 v = a_1\lambda_1 A v_1 + a_2\lambda_2 A v_2 + \cdots a_n\lambda_n A v_n \qquad \textbf{(1.31)}$$

En consecuencia, tendríamos:

$$A^2 v = a_1{\lambda_1}^2 v_1 + a_2{\lambda_2}^2 v_2 + \cdots a_n{\lambda_n}^2 v_n \qquad \textbf{(1.32)}$$

Si continuamos multiplicando por A, de forma sucesiva tenemos:

$$A^k v = a_1{\lambda_1}^k v_1 + a_2{\lambda_2}^k v_2 + \cdots a_n{\lambda_n}^k v_n \qquad \textbf{(1.33)}$$

Como tenemos que le valor propio λ_1 es de mayor modulo, si sacamos factor común ${\lambda_1}^k$ en la expresión anterior tenemos:

$$A^k v = {\lambda_1}^k \left[a_1 v_1 + a_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k v_2 + \cdots + a_n \left(\frac{\lambda_n}{\lambda_1}\right)^k v_n \right] \qquad \textbf{(1.34)}$$

Si tomamos límite cuando $k \to \infty$, y tenemos en cuenta que al ser λ_1 el valor propio dominante entonces:

(1.35)

$$\lim_{k\to\infty}\left(\frac{\lambda_i}{\lambda_1}\right)^k = 0 \qquad\qquad i = 2,3,\dots n$$

Entonces en consecuencia se tiene que:

$$\lim_{k\to\infty}\left[a_1v_1 + a_2\left(\frac{\lambda_2}{\lambda_1}\right)^k v_2 + \cdots + a_n\left(\frac{\lambda_n}{\lambda_1}\right)^k v_n\right] = a_1v_1 \qquad \textbf{(1.36)}$$

En consecuencia, para valores grandes de k se tiene que:

$$A^k v \approx {\lambda_1}^k a_1 v_1 \qquad \textbf{(1.37)}$$

Es decir, la sucesión:

$$\left\{{\lambda_1}^{-k} A^k v\right\}_{k\in N} \qquad Converge\ a\ \ a_1v_1 \qquad \textbf{(1.38)}$$

Por lo tanto, tenemos que:

$$\lim_{k\to\infty}\frac{(A^{k+1}v)_r}{(A^k v)_r} = \lambda_1 \quad r\colon\ r - esimo\ componente \qquad \textbf{(1.39)}$$

Es decir, la esencia del método de la potencia es multiplicar la estimación inicial por una potencia de A normalizada adecuadamente.

1.2. WEB CRAWLER

Dada la cantidad de información disponible en la web, es imposible indexar de forma manual, como ocurre a menudo en las bibliotecas, por lo que este proceso debe hacerse de una forma automática y sistemática. El agente responsable de esta tarea por lo general se llama Web Crawler. Las definiciones aquí citadas se encuentran en [23] y [24].

1.2.1. DEFINICION

Un Web Crawler (o Araña Web) es un programa que inspecciona las páginas del World Wide Web de forma metódica y automatizada. Uno de sus usos más frecuentes consiste en crear una copia de todas las páginas web visitadas para su procesado posterior por un motor de búsqueda que indexa las páginas proporcionando un sistema de búsquedas rápido.

Los webs crawlers comienzan visitando una lista de URLs, identifican los enlaces en dichas páginas y los añaden a la lista de urls a visitar de manera recurrente de acuerdo a determinado conjunto de reglas. El cómputo usual de un Crawler es a partir de un grupo de direcciones iniciales (semillas) cuyos recursos apuntados son descargados y analizados en busca de enlazar a nuevos recursos, por lo general páginas html, repitiendo estos procesos hasta que las condiciones finales sean alcanzadas. Estas condiciones varían de acuerdo a la política de Crawling deseada.

1.2.2. BREVE HISTORIA DE LOS WEB CRAWLER

El primer Web Crawler fue muy simple, principalmente por la escala y el contenido de la Web, ya que, a finales de 1993, principios de 1994 Internet era pequeño, limitado principalmente para investigación e instituciones educativas.

De hecho, el índice del primer Web Crawler contenía páginas de sólo 6.000 sitios. Tenía cuatro piezas centrales; una base de datos en la que almacenar el estado, un conjunto de agentes que recupera páginas de la Web, un operador para coordinar los agentes y la base de datos y un índice para actualizar los índices de textos completos con nuevos documentos recuperados. Su arquitectura se puede apreciar en la figura 1.3.

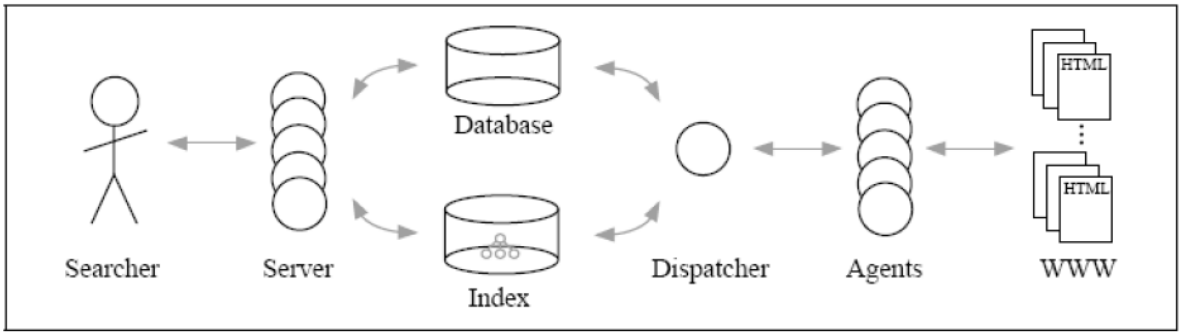


Figura 1. 3. Arquitectura del primer Web Crawler
Fuente: [24]

En 1994 Eichman publica el primer crawler, RBSE, cuya arquitectura se basa en dos componentes: "*spider*", que actualiza la base de datos y "*mite*", que descarga los documentos y los almacena como documentos ASCII.

En enero de 1994 Brian Pinkerton, un estudiante de Informática de la universidad de Washington, comenzó un Web Crawler en su tiempo libre, *World Wide Web Worm*, que es el primer índice público de títulos y urls.

En 1997 Burner crea el Internet Archive, que guarda un histórico de hosts e IPs. En 1998 aparece WebSPHINX (Miller), que es el primero en utilizar multi-hilo y HTML parking. En ese mismo año, Bring y Page crean Google Crawler, donde el crawler y el sistema están integrados, utilizan una arquitectura centralizada.

1.2.3. FUNCIONAMIENTO DE LOS WEBS CRAWLER

Típicamente, una página web contiene referencia a otras páginas, las que, en general, apuntan al mismo sitio, es decir que una página contiene referencias a subpáginas del mismo sitio. Sin embargo, cuando una referencia direcciona a otro sitio, entonces es posible "*saltar*" a este nuevo sitio, de tal manera que se puede ir navegando dentro de la web.

En general, un crawler dispone de un conjunto inicial de urls, conocidas como semilla, entonces el crawler va descargando las páginas web asociadas a las semillas y buscando dentro de éstas otras urls. Cada url nueva encontrada se añade a la lista de urls que el crawler debe visitar. A este proceso se le denomina recolección de urls. Ver figura 1.4.

El crawler accede a una nueva url, la página web asociada es descargada al computador local. Una vez ahí, éstas son parseadas y procesadas. Es importante mencionar que ningún crawler puede acceder a todas las urls que hay en Internet, pues el número de páginas existentes es gigantesco. Estadísticamente, el porcentaje de Internet que suele ser explorado por un crawler es aproximadamente del 15%.

Tras el parseo, el crawler procesa la información disponible, es decir, aplica algún tipo de algoritmo para conseguir el objetivo establecido. Por ejemplo, comprobar la disponibilidad de un enlace o verificar el tamaño de una imagen.

Es posible que se establezca un nivel de profundidad en un sitio, siendo este el número de sublinks que ha de atravesar para llegar al sublink más lejano del link principal. Una vez que está en cada página web, almacena los documentos para su posterior tratamiento o directamente, a través de algún algoritmo extrae la información y la introduce en alguna base de datos.

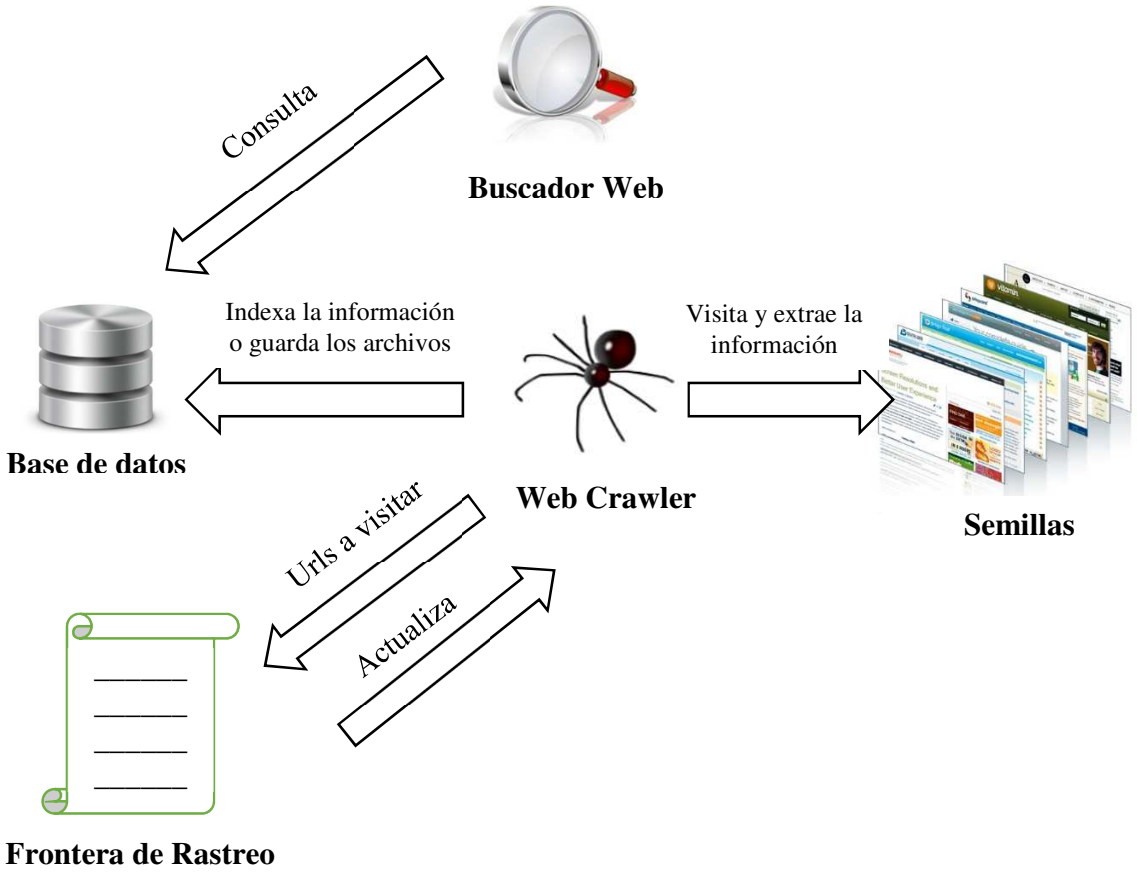


Figura 1. 4. Funcionamiento de un Web Crawler

1.2.4. USO DE LOS WEB CRAWLERS

Uno de los usos más frecuentes que se les da, consiste en crear una copia de todas las páginas web visitadas, para su posterior proceso en un motor de búsqueda, que indexa las páginas proporcionando un sistema de búsquedas rápido.

Su principal uso es en motores de búsqueda, pero éste no es el único uso en absoluto. Las arañas Web (o crawlers) pueden utilizarse para multitud de tareas. Por ejemplo:

- Invitar a todos los miembros de una red social, que participen en una manifestación pública.
- Verificar que todos los enlaces de una página Web apuntan a páginas activas.
- Crear periódicamente un informe de cambios del sitio de una empresa competidora
- Enviar un mensaje de correo electrónico cuando se añadan noticias a una página de noticias.
- Realizar un estudio estadístico sobre el uso de un determinado término en páginas latinas.

El uso de un crawler puede ser muy variado, estando los límites establecidos únicamente en la imaginación.

1.3. INTRODUCCION AL ENTORNO DE OPENWEBSPIDER(JS)V.0.2.3

Software con licencia OpenSource y de libre distribución.

1.3.1. REQUERIMIENTOS

Antes de implementar el OpenWebSpider, debemos de tener instalado los siguientes elementos:

- Node.js, tecnología que permite trabajar con Javascript del lado del servidor. Disponible en *https://nodejs.org/en/download/*

- Gestor de base de datos Mysql Server 5.5 o superior. Disponible en *https://dev.mysql.com/downloads/mysql/*

1.3.2. CONFIGURACION

Una vez instalado los componentes mencionados, se procederá a realizar las siguientes acciones:

- Descargar desde la página oficinal el OpenWebSpider (js) versión 0.2.3 disponible en *https://sourceforge.net/projects/openwebspider/*

- Descomprimir el archivo "*openwebspider_js_0.2.3.zip*", dentro de la carpeta descomprimida, observara una serie de archivos. Al ejecutar el archivo "*openwebspider.bat*", se iniciará la aplicación en modo cosola, como la que se muestra en la figura 1.5.

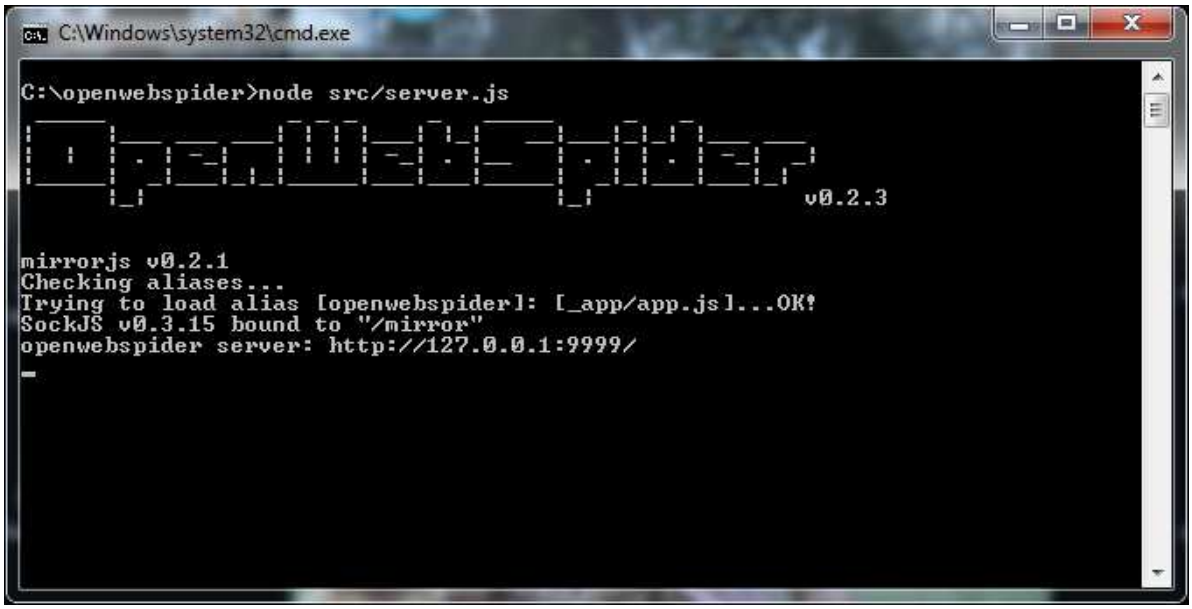


Figura 1. 5. Ventana inicial de la aplicación
Fuente: sourceforge.net, 2016

- Abra el navegador de su elección y en la barra de direcciones digitar y ejecutar la dirección *http://127.0.0.1:9999/* . (figura 1.6)

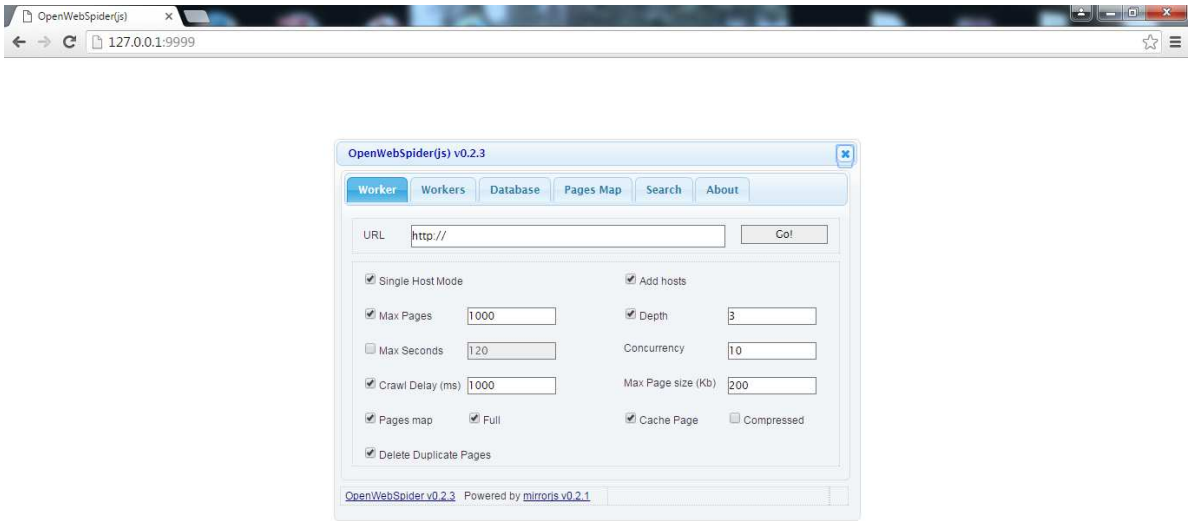


Figura 1. 6. OpenWebSpider(js) v0.23
Fuente: sourceforge.net, 2016

1.3.3. CONFIGURACION DE LA BASE DE DATOS

En primera instancia se configurará la base de datos en OpenWebSpider(js) v0.23, para tal fin seleccionamos la pestaña "*Database*" que se muestra en la figura 1.7, se mostraran todas las opciones de configuración que a continuación detallamos en la tabla 1.1.

Tabla 1. 1. Argumentos y acciones para configurar la base de datos en OpenWebSpider(js).

Argumentos	Acción
Database Type	Seleccionar el Gestor de base de datos instalado, para este caso MySQL.
Host	Referencia al servidor virtual con que nos conectaremos, para ello digitamos en la caja de texto correspondiente la dirección 127.0.0.1
Port	Puerto de enlace entre el servidor virtual y el gestor de base de datos MySQL, que por defecto se inserta en la caja de texto correspondiente un valor numérico que es configurado al momento de la instalación del MySQL, para este caso 3306.
Username	Usuario para ingresar al gestor de base datos, por defecto se inserta en la caja de texto correspondiente el nombre de *root*.
Password	Clave para ingresar al gestor de base de datos, esto es configurado al momento de la instalación del MySQL.
Host DB	Base de datos en donde se insertarán todos las páginas rastreadas y sus relaciones existentes, esta base de datos es *ows_hosts*, que por defecto se inserta en la caja de texto correspondiente.
Index DB	Base de datos donde se insertarán todas las páginas que son indexadas, esta base de datos es *ows_index* que por defecto se inserta en la caja de texto correspondiente.

Figura 1. 7. Opciones para la configuración de la base de datos.
Fuente: Sourceforge.net, 2016

Configuradas las variables correspondientes, verificaremos que OpenWebSpider(js) v0.23 conecte correctamente a su servidor, para tal fin se hace clic en el botón "*Verify*". Si es conforme, en la parte inferior derecha mostrara el mensaje "*Connection OK*", confirmando que la conexión ha sido satisfactoria. Inmediatamente guardamos la configuración haciendo clic en el botón "*Save*". Ver figura 1.8

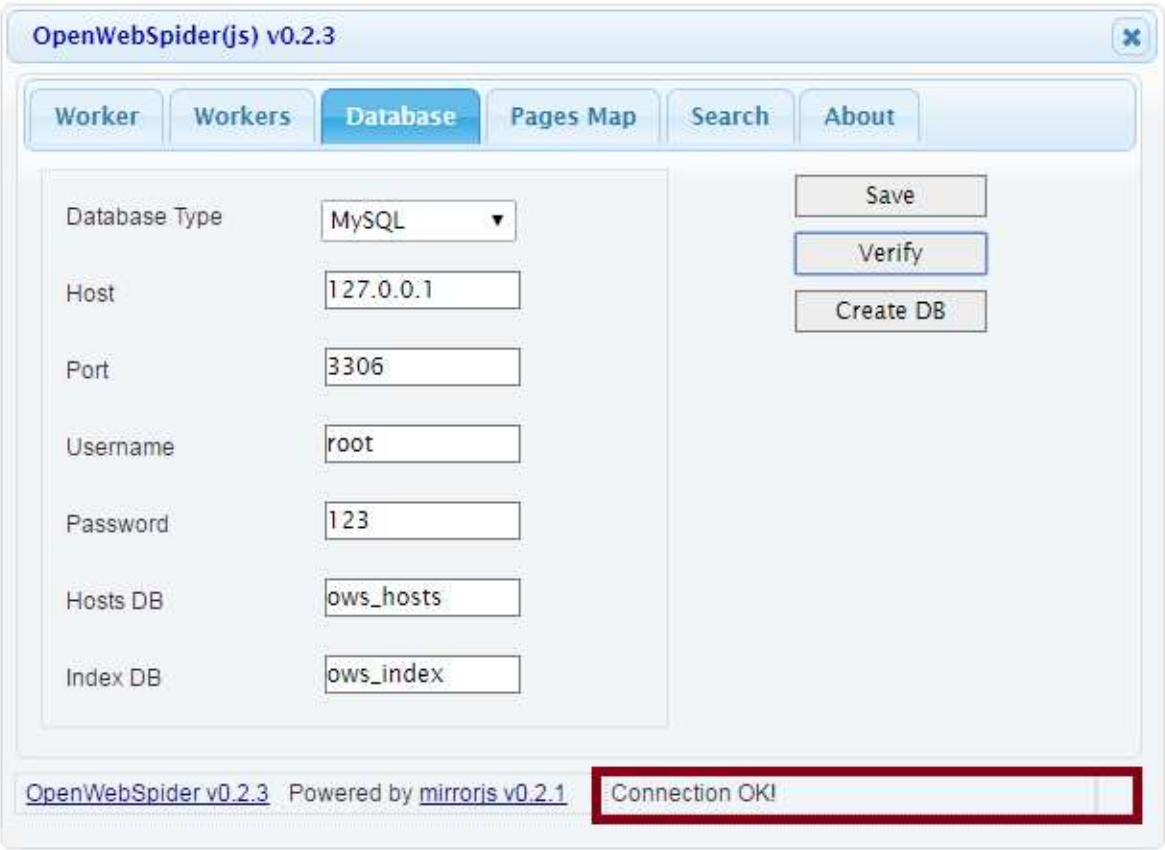


Figura 1. 8. Opciones para la configuración de la base de datos/conexión establecida.

Comprobada la conexión, hacemos clic en el botón "Create DB", esto generara las bases de datos y las tablas necesarias para el correcto funcionamiento de Crawler. Al finalizar, en la parte inferior derecha, mostrará el mensaje "*Database created/truncated/Repaired*", indicándonos que todo fue satisfactorio. Ver figura 1.9.

Figura 1. 9. Creación de la base de datos y sus respectivas tablas.

Se comprobará la creación de las bases de datos *ows_hosts* y *ows_index*, en MySQL Workbench, como se muestra en la figura 1.10:

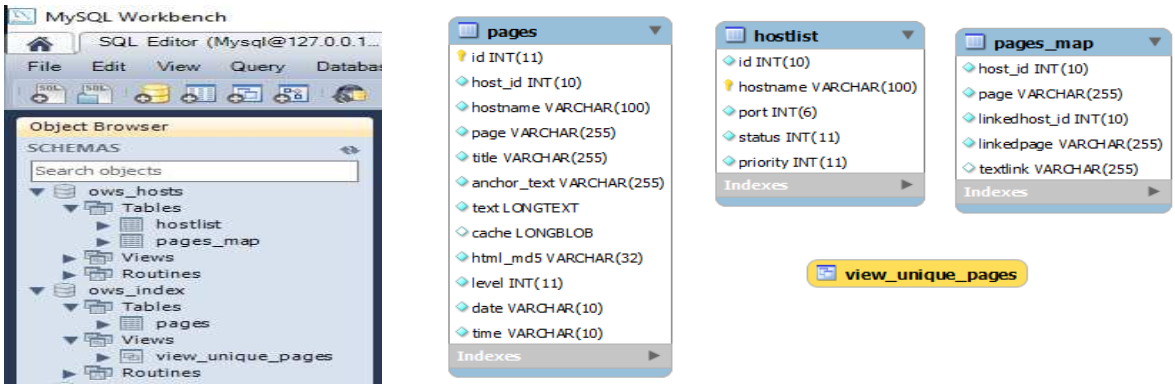


Figura 1. 10. Vista de las bases de datos y sus respectivas las tablas.

1.3.4. RASTEAR URLS

En seguidamente se rastreará las urls que se asignen, para tal fin, procedemos con las siguientes acciones:

Activamos, la ficha "*Worker*", seguidamente mostrará la figura 1.11, con la cual se mostrarán las opciones de configuración que se detallan en la tabla 1.2.

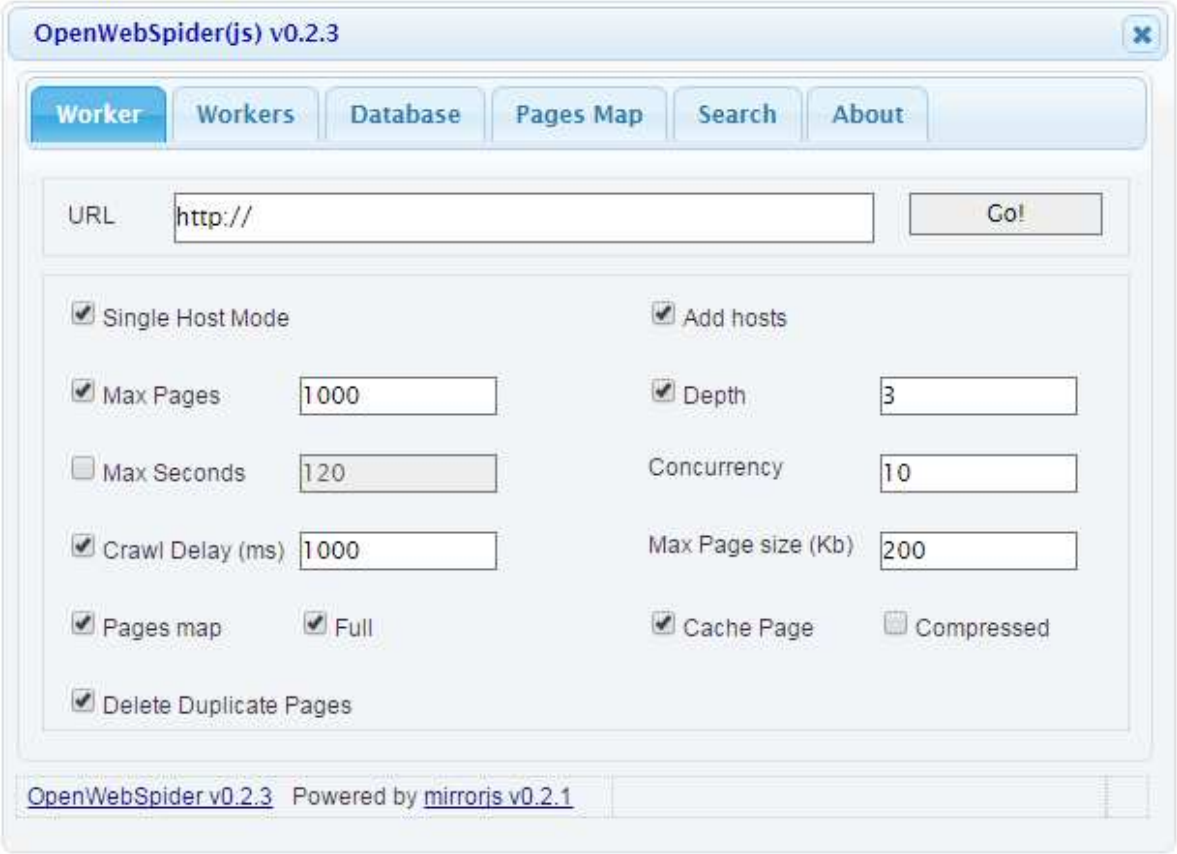


Figura 1. 11. Ventana principal para rastrear Urls.

Tabla 1. 2 . Argumentos y acciones para rastrear urls en OpenWebSpider(js).

Argumento	Objeto	Acción
Single Host Mode	Casilla de selección	Este argumento debe utilizarse cuando se desea indexar sólo un sitio web. Si omite este argumento, el WebCrawler indexará la URL especificada y recursivamente todos los sitios web enlazados por la anterior
Max Page	Casilla de selección y caja de texto	Asigna el número máximo de páginas a indexar por dominio
Max Seconds	Casilla de selección y caja de texto	Asigna un máximo de segundos por dominio
Crawl Delay (ms)	Casilla de selección y caja de texto	Milisegundos entre la descarga de una página y la siguiente.
Page Map	Casilla de selección	Lista las páginas de un sitio web accesibles por parte de los buscadores.
Delete Duplicate Pages	Casilla de selección	Elimina las páginas duplicadas.
Add hosts	Casilla de selección	Añade el nombre del host a la lista de hosts, imprime el Id y sale.
Depth	Casilla de selección y caja de texto	Asigna el máximo nivel de profundidad (Por defecto, indexa todas las páginas).
Concurrency	Casilla de selección y caja de texto	Para evitar que distintos threads (subproceso) procesen las mismas páginas, e intenten guardarlas en el mismo archivo local.
Max Page Size(KB)	Casilla de selección	Asigna un máximo de Kb de descarga por página.
Cache	Casilla de selección	Guarda una copia de cada página indexada
Compressed	Casilla de selección	Guarda una copia comprimida de cada página indexada
URL	Caja de texto	Url que se desee rastrear.
GO!	Botón	Empieza con el rastreo.

A continuación, digitamos en la caja de texto "*URL*", la url a rastrear, para este caso será *www.unapiquitos.edu.pe* , si fuera necesario alteraremos los argumentos (para esta tesis lo dejaremos tal como muestra la figura 1.7), finalmente presionamos el botón "*GO!*". Ver figura 1.12.

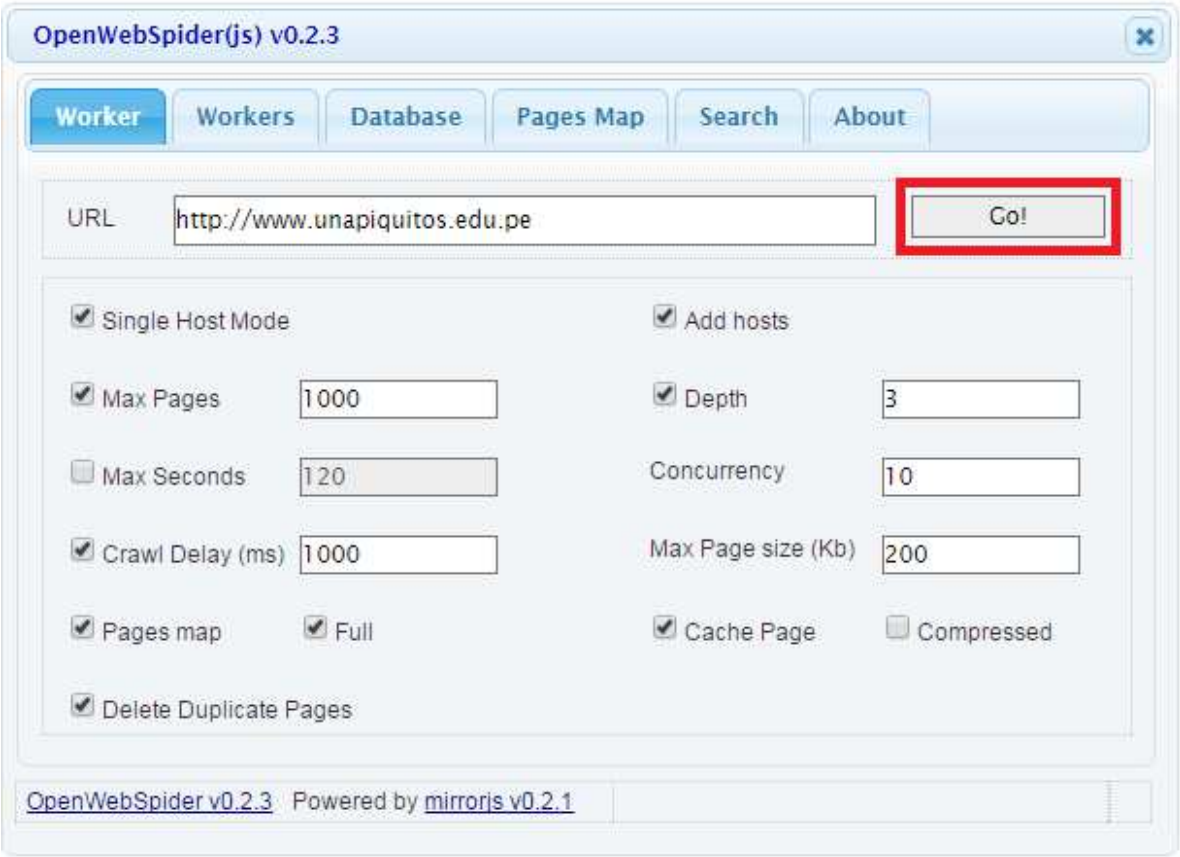


Figura 1. 12. Ventana para rastrear la Url www.unapiquitos.edu.pe.

Seguidamente, se volcará a la pestaña "*Workers*". Aquí observamos la url que se está rastreando, cantidad de páginas indexar, concurrencia, estado, etc, además el progreso en tiempo real. Ver figura 1.13.

Figura 1. 13. Proceso de rastreo de la url asignada www.unapiquitos.edu.pe.

Finalizado el rastreo de la url, observamos que los datos fueron insertados por el OpenWebSpider(js), en la base de datos "*ows_host*" y "*ows_index*" en Mysql Workbench como muestra la figura 1.14.

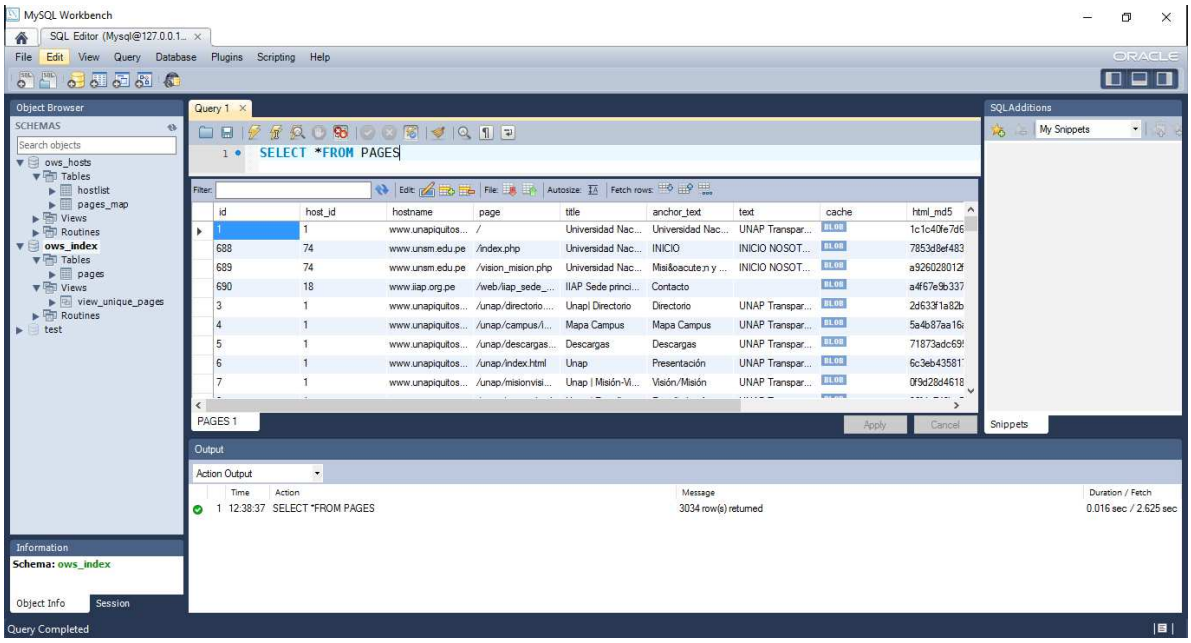


Figura 1. 14. Vista de la Tabla *pages* de la base de datos *ows_index*.

1.4. INTRODUCCION AL ENTORNO DE PAJEK

1.4.1. DEFINICIÓN

El programa computacional Pajek (*Pajek* significa araña en esloveno), es un software gratuito (no libre) para usos no comerciales desarrollado para el análisis y visualización de grandes redes, creado en la universidad de Ljubljana, Eslovenia, por Vladimir Batagelj (a la derecha en la figura 1.15) (matemático), Andrej Mrvar (a la izquierda en la figura 1.15) (sociólogo) y la contribución de Matjaž Zaveršnik [25]. Se empezó a desarrollar en noviembre 1996 y se implementó en Delphi (Pascal).

Figura 1. 15 Andrej Mrvar y Vladimir Batagelj
Fuente: [26]

Funciona sobre plataformas Windows y es muy usado dentro del entorno de análisis de redes sociales. Puede descargarse del sitio *http://mrvar.fdv.uni-lj.si/pajek/*.

Algunas de las redes que pueden ser analizados y visualizados con Pajek Son los siguientes [25]:

- Redes de moléculas orgánicas en la química orgánica
- La proteína Red de Interacción - receptores
- Las redes sociales

- árboles genealógicos
- Redes de Internet
- redes de citación
- Las redes de difusión (virus, noticias, etc.)

1.4.2. INICIANDO PAJEK

Una vez que se haya descargado la aplicación portable o instalable, hay que ejecutarlo de la manera convencional desde el directorio que seleccionemos. En seguidamente se mostrará la figura 1.16. Las definiciones aquí citadas se encuentran en [27] y [28]

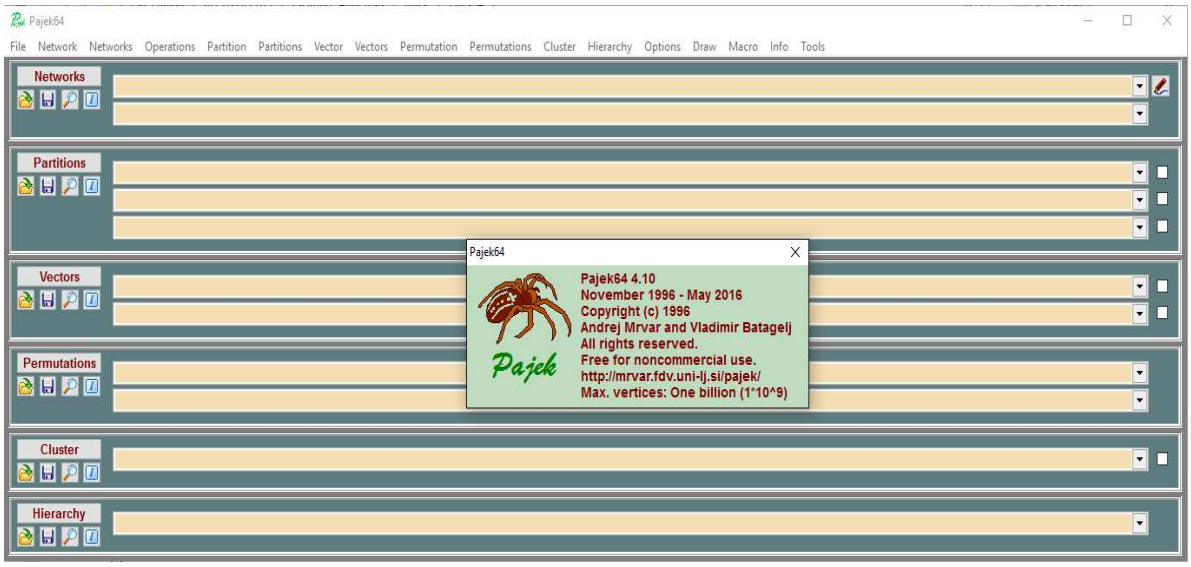


Figura 1. 16. Ventana principal de Pajek
Fuente: Vladimir Batagelj, Andrej Mrvar

Como observamos en la figura 1.16, tenemos 6 secciones con los que podríamos crear nuestro grafo. Tenemos la opción "*Network*" que es donde añadiremos la red que creemos, "*Partitions*" donde insertaremos las particiones y diferenciaciones de ciertas partes del grafo que introduzcamos, "*Vectors*" donde introduciremos la forma de los nodos para su diferenciación según la importancia, "*Permutations*" que son los datos de los nodos y como serian según esa información, "*Cluste*r" donde podremos crear los grupos del grafo y finalmente "*Hierarchy*" la cual nos ayudara a crear las jerarquías que queramos introducir en nuestro grafo.

1.4.3. GENERANDO GRAFOS EN PAJEK

Los grafos en Pajek se diferencian en vértices que son los actores, es decir los nodos de nuestro grafo y las relaciones que los enlazan, que en Pajek son Edges (no dirigidos) o Arcs (dirigidos). Las definiciones aquí citadas se encuentran en [27] y [28]

La definición de los actores empieza en la primera línea con el identificador "**Vertices*". La definición de los vínculos va línea seguido de la definición de los actores y empieza con el identificador "**Edges*" o "**Arcs"*.

En el identificador *Vertices se declara el número de actores dejando un espacio de separación. En la siguiente línea y uno por línea se le asigna un número a cada actor y se escribe el nombre entre comillas (″) dejando un espacio de separación. En el identificador **Arcs*(deseamos que las enlaces sea dirigidos), para cada vínculo se usará el número que se le asignó a cada actor y se declara uno por línea, así para definir el vínculo entre la url *www.unapiquitos.edu.pe* y *www.iiap.org.pe* se escribirán los números 1 y 2 separados por un espacio. Estas acciones se inician generando un archivo txt, para luego digitar los vértices y edges o arcs, como se muestra en la figura 1.17.

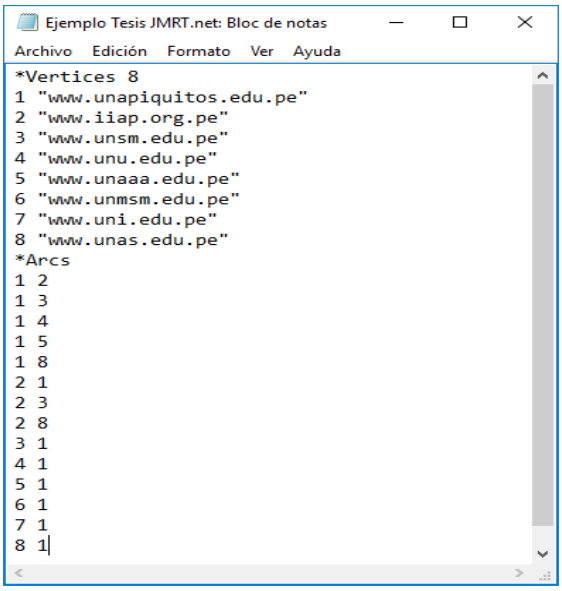


Figura 1. 17. Creación del archivo txt para declarar los vértices y los arcos.

Finalizado la declaración de nuestros nodos (vértices) y las relaciones que existen entre ellos (edges o arcs), se guardará el archivo con extensión .net, para esta tesis el archivo tendrá el nombre de "*Ejemplo Tesis JMRT.net*".

A continuación, deberemos abrir, este archivo en Pajek. Ir al menú "*File*", posteriormente a "*Network*" y finalmente a "*Read*" para elegir el archivo "*Ejemplo Tesis JMRT.net"* y seleccionarlo. Automáticamente nos abrirá una ventana de "*Report*" donde se irán guardando el resultado de los procesos y si hubiera algún error lo indicara. Ver figura 1.18 y 1.19.

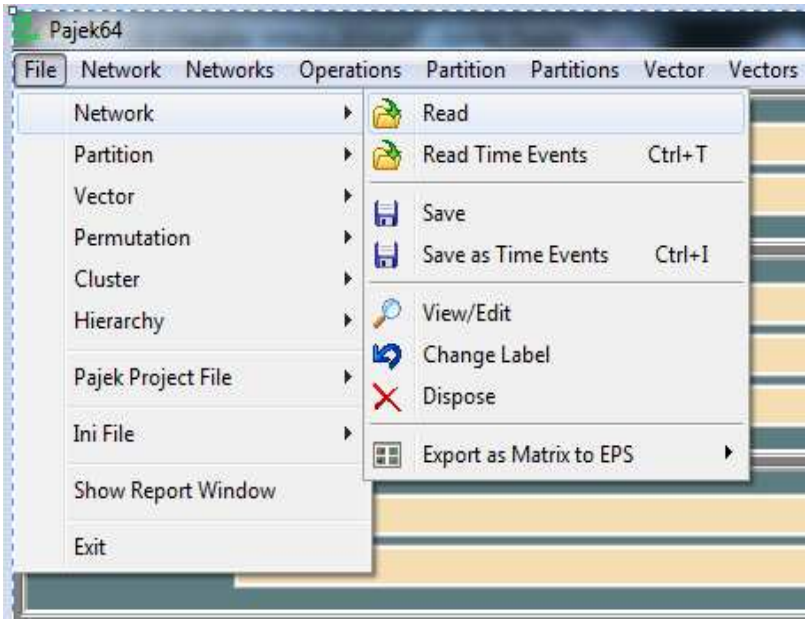


Figura 1. 18 Abrir archivos .NET en Pajek.

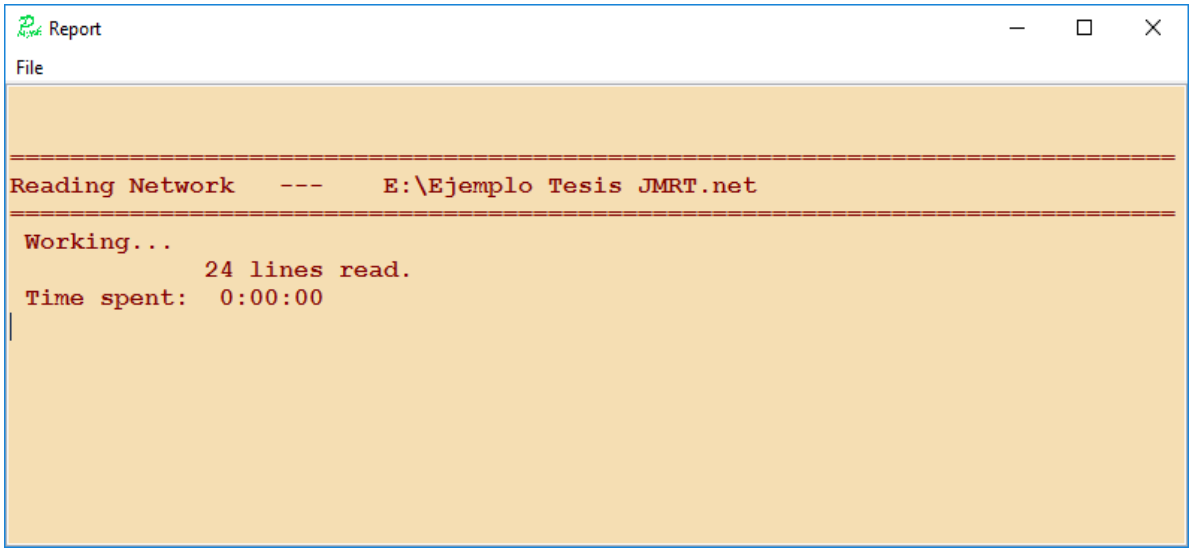


Figura 1. 19. Reporte del archivo .NET seleccionado.

Una vez importado el archivo .NET, para observar el grafo que representa deberemos ir a la opción "*Draw*" y pulsar en la opción "*Network*". Así obtendremos nuestro grafo del archivo importado. Ver figura 1.20 y 1.21.

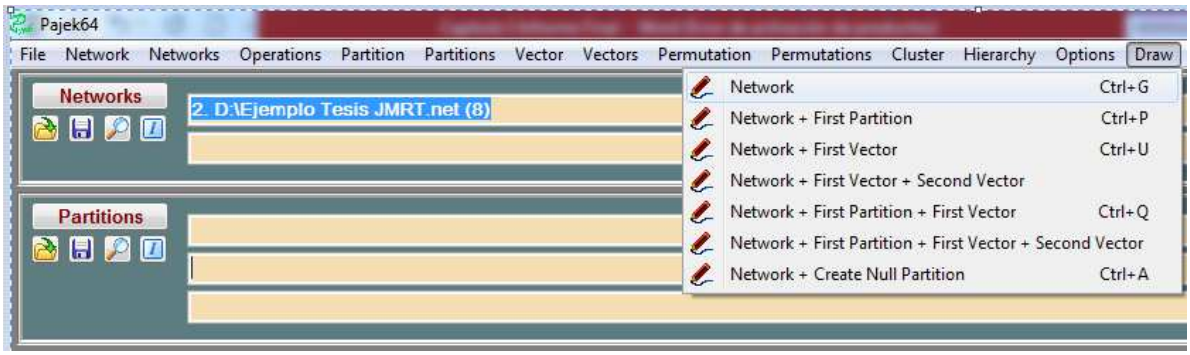


Figura 1. 20. Acciones para visualizar el grafo generado en Pajek.

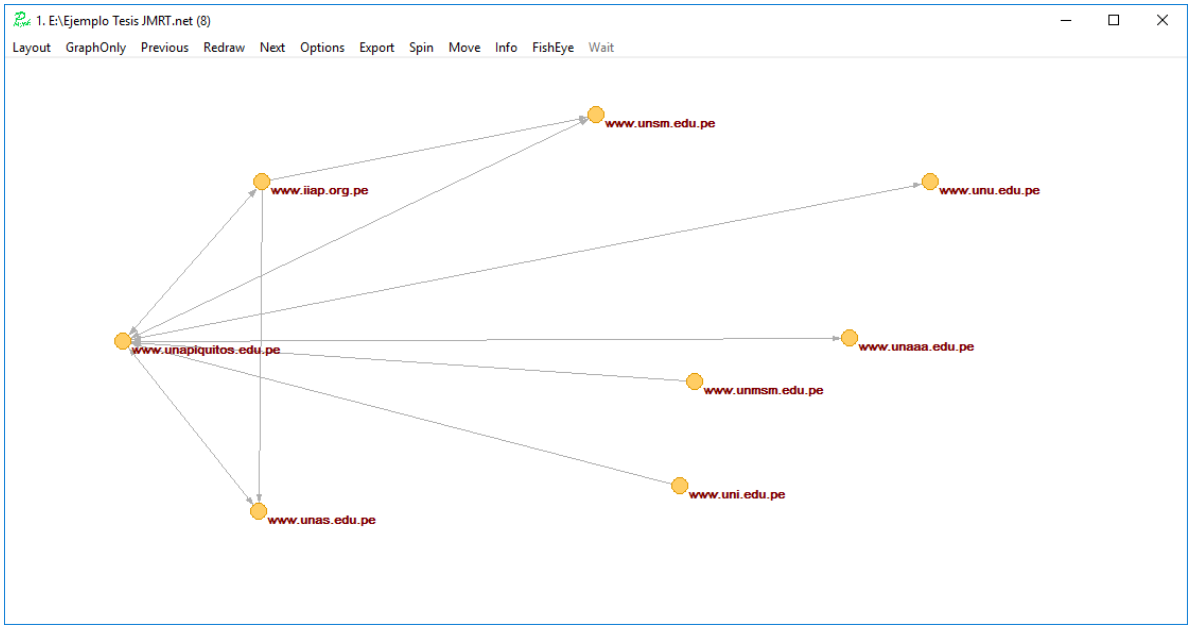


Figura 1. 21.Grafo del archivo "*Ejemplo Tesis JMRT.net*" en Pajek.

1.4.4. CENTRALIDAD

La centralidad es un concepto clave en el análisis de redes sociales. Se utiliza comúnmente para medir la importancia o el poder de un nodo. El concepto se basa en que la posición de un nodo afecta a su influencia a los otros nodos en las redes.

Para generar un grafo con centralidad, sin salirnos de Pajek debemos ir al menú "*Network*", posteriormente a "*Create Vector*", "*Centrality*", "*Weighted Degree*" y finalmente a "*All*". Ver figura 1.22.

Para observar el grafo que representa deberemos ir a la opción "*Draw*" y pulsar en la opción "*Network + First Vector*". Así obtendremos nuestro grafo con centralidad. Ver figura 1.23.

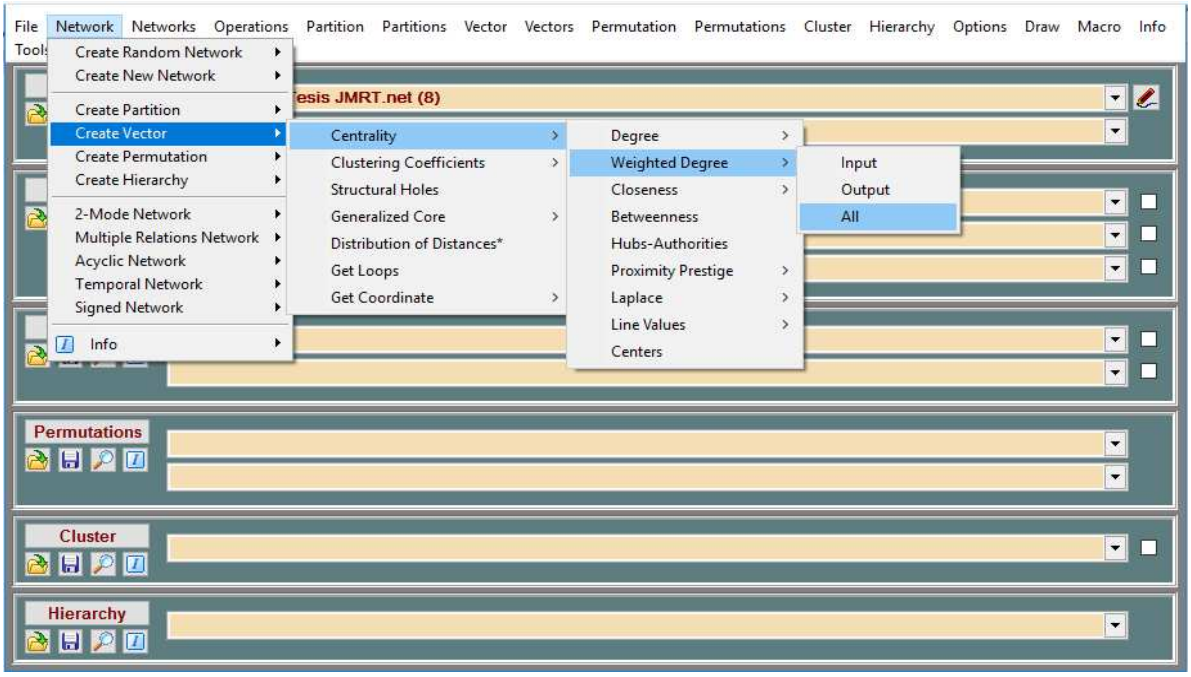


Figura 1. 22. Acciones para generar un grafo con centralidad

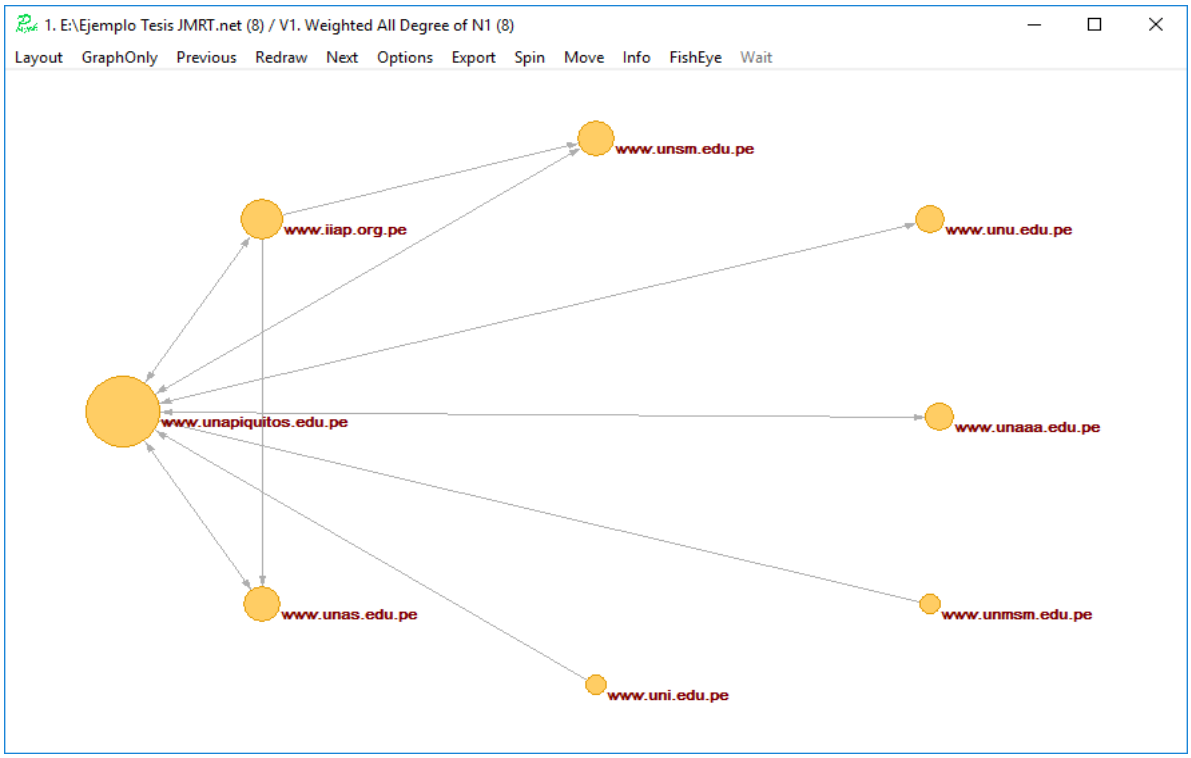


Figura 1. 23. Grafo con centralidad del archivo “Ejemplo Tesis JMRT.net”

Podemos realizar diferenciaciones entre los nodos y las relaciones. Por ejemplo, podemos diferenciar los nodos con diferentes colores, reducir los nodos que no tienen enlaces entrantes ni salientes, dependiendo de que si queremos dar más visibilidad a nuestro grafo de conexiones.

Para ello habría que modificar nuestro código del archivo "*Ejemplo Tesis JMRT.net"* o realizarlo por medio de las opciones que nos provee el Pajek.

1.5. INTRODUCCION AL MATLAB

En esta sección se pretende realizar una introducción muy básica a MATLAB, orientado fundamentalmente al estudio del algebra lineal. MATLAB constituye un entorno abierto para el cual numerosos paquetes específicos adicionales(toolboxes) han sido desarrollados, estos paquetes específicos adicionales están constituidos por un conjunto de funciones que pueden realizar multitud de análisis. Las Notas se centrarán fundamentalmente en aquellos aspectos y funciones que más interés tenga para esta tesis, desde el punto de vista del algebra lineal. Las definiciones aquí citadas se encuentran en [29], [30], [31], [32], [33] y [34]

1.5.1. DEFICION

MATLAB (abreviatura de MATrix LABoratory, "laboratorio de matrices") es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows, Mac OS X y GNU/Linux .

Entre sus prestaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware. El paquete MATLAB dispone de dos herramientas adicionales que expanden sus prestaciones, a saber, Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario - GUI). Además, se pueden ampliar las capacidades de MATLAB con las *cajas de herramientas* (*toolboxes*); y las de Simulink con los *paquetes de bloques* (*blocksets*).

Es un software muy usado en universidades y centros de investigación y desarrollo. En los últimos años ha aumentado el número de prestaciones, como la de programar directamente procesadores digitales de señal o crear código VHDL.

En líneas generales, MATLAB en un sistema interactivo basado en matrices para cálculos científicos y de ingeniería. MATLAB se puede considerar un entorno matemático de simulación que puede utilizarse para modelar y analizar sistemas, sirve para estudiar sistemas continuos, discretos, lineales y no lineales.

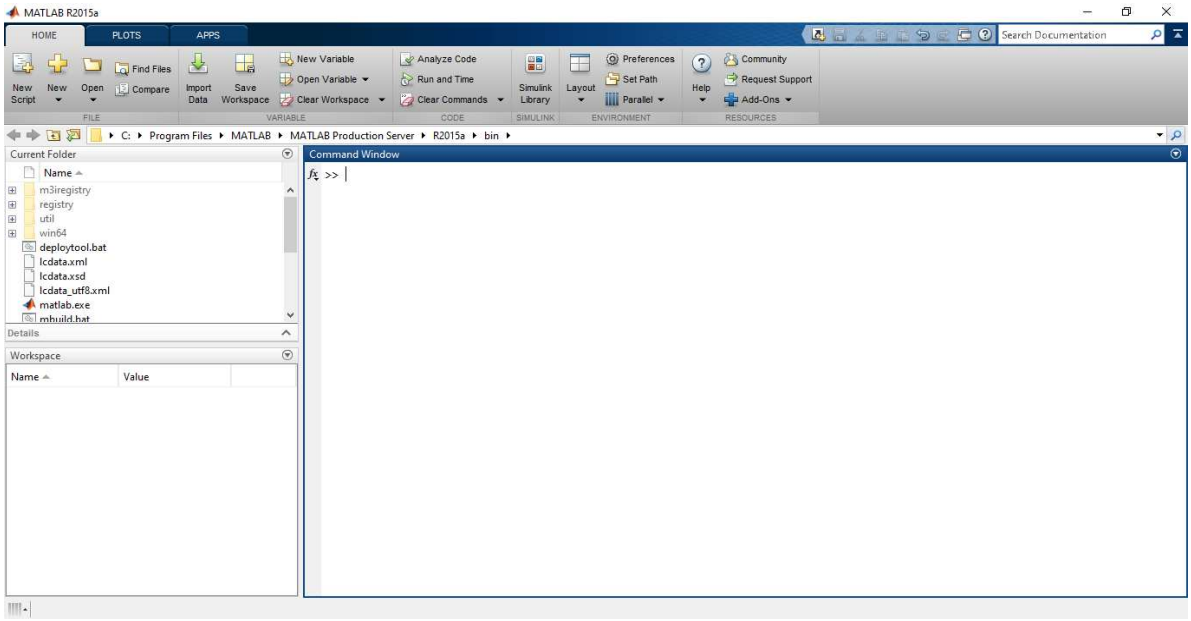


Figura 1. 24. Ventana principal de Matlab R2015a
Fuente: Mathwork

1.5.2. ENTORNO

Una vez iniciado MATLAB, aparece el prompt o línea de comandos del sistema ($fx \gg$) , que indica que el programa está listo y esperando comandos. Ver figura 1.25.

Figura 1. 25. Línea de comandos del sistema.
Las demás ventanas que forman parte del entorno de trabajo de MATLAB son:

Command history. Las órdenes introducidas en la ventana command window quedan grabadas en esta ventana, de forma que, haciendo doble click sobre ellas, las podemos volver a ejecutar. De la misma forma, si nos situamos en la ventana command window, en el espacio reservado para introducir nuevas órdenes, inmediatamente después de >>, y presionamos la tecla ↑, podemos acceder a órdenes introducidas con anterioridad. Ver figura 1.26.

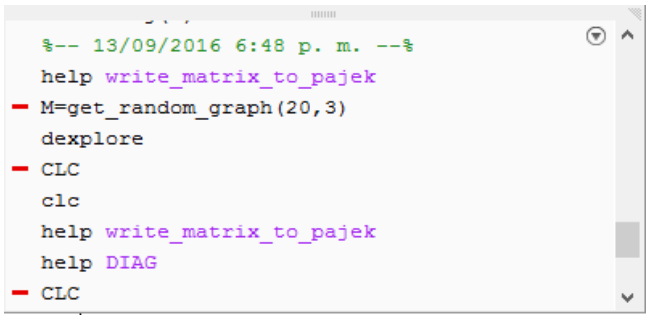


Figura 1. 26 Ventana Command history.

Workspace. Esta ventana contiene las variables (escalares, vectores, matrices, etc.) creadas en la sesión de MATLAB. La ventana workspace nos proporciona información sobre el nombre, dimensiones, tamaño y tipo de variable (Figura 1.27). Existen dos opciones para eliminar una variable:

- Introducir en command window el comando clear seguido del nombre de la variable.
- Seleccionar la variable en el workspace y borrarla directamente con la tecla delete.

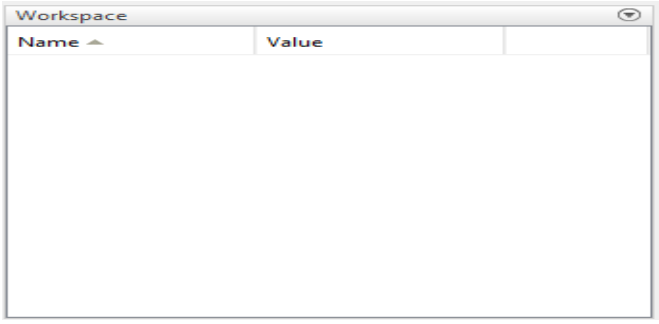


Figura 1. 27 Ventana Workspace

Current Folder. Las operaciones de MATLAB utilizan el directorio seleccionado en current directory (a través del botón para explorar) como punto de referencia.

Por ejemplo, si guardamos una serie de variables con el comando save, se guardan en el directorio en el que estemos trabajando. Lo mismo ocurre cuando cargamos datos con el comando load: el ordenador busca los datos en el fichero en el que estemos trabajando.

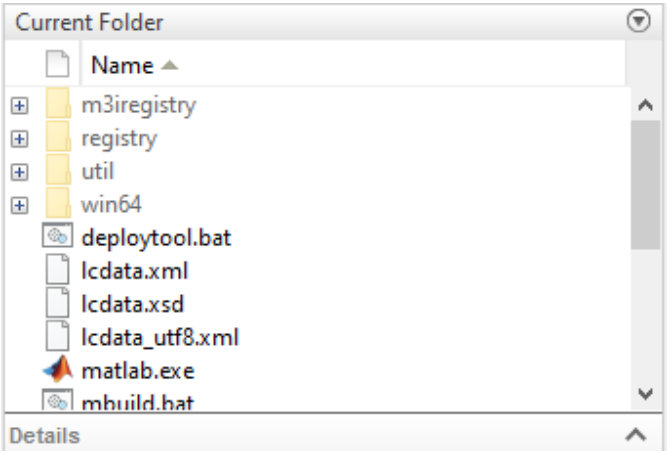


Figura 1. 28.Ventana Current Folder.

1.5.3. OPERADORES BASICOS

La Forma más simple de usar MATLAB es como una calculadora, en lugar de pulsar teclas se ha de escribir la operación a realizar y pulsar ENTER para que esta se lleve a cabo. Las operaciones disponibles son: + (suma), - (resta), * (producto), / (división) y ˆ (exponenciación).

Por ejemplo, tecleando:

$fx \gg$ 10+5

MATLAB responderá con el resultado de la operación:

ans =15

1.5.4. DECLARACIÓN DE VARIABLES

Una variable es un espacio en memoria al cual se le asigna una etiqueta o nombre que lo caracteriza y un contenido, La etiqueta puede tener hasta 63 caracteres y debe de empezar con una letra. El contenido puede ser numérico, lógico carácter, es conveniente que el nombre de las variables esté relacionado con sus contenidos para evitar confusiones.

En líneas generales, las reglas para nombrar una variable son:

- El nombre de una variable puede tener como máximo 63 caracteres, que pueden ser letras, Alfanúmeros y el guion de subrayar.
- El primer carácter tiene que ser una letra, vector1 es un nombre valido, pero no lo es 1vector.
- Las mayúsculas y las minúsculas tienen un valor distintivo. La variable Vector es distinta a la variable vector.
- Dentro de un nombre de variables no puede haber espacios en blanco, vector1 es válido, pero no lo es vector 1

Por ejemplo, tecleando:

$fx \gg$ A=10
MATLAB responderá con el resultado:
A =10
$fx \gg$ B=5
MATLAB responderá con el resultado:
B =5
$fx \gg$ A+B
MATLAB responderá con el resultado de la operación:
ans =15

1.5.5. MATRICES Y VECTORES

En MATLAB todas las variables son matrices; en particular, los escalares son matrices 1×1. En realidad, MATLAB proviene de "MATrix LABoratory".

Las matrices se indican por corchetes, los elementos de una misma fila se separan mediante "," (o también espacios) y las filas se separan con ";" (o saltos de línea). Por ejemplo, al teclear:

$fx \gg A = [2\ 1\ 0; 7\ 4\ 1; 4\ 5\ 8]$

Obtenemos la matriz A

$$A = \begin{matrix} 2 & 1 & 0 \\ 7 & 1 & 4 \\ 4 & 5 & 8 \end{matrix}$$

Los vectores son matrices de una sola fila (vectores fila) o columna (vectores columna). Por ejemplo.

Tabla 1. 3. Vector fila y vector columna en MATLAB

Vector fila	Vector columna
$fx \gg B = [7\ 3\ 4\ 9\ 2\ 11]$ Obtenemos el vector B $B = 7 \quad 3 \quad 4 \quad 9 \quad 2 \quad 11$	$fx \gg B = [7; 3; 4; 9; 2; 11]$ Obtenemos el vector B $B = \begin{matrix} 7 \\ 3 \\ 4 \\ 9 \\ 2 \\ 11 \end{matrix}$

MATLAB puede operar con matrices por medio de operadores y por medio de funciones. Tomemos como ejemplo las matrices A y B(sus elementos serán generados en el primer ejemplo de la tabla), seguidamente se mostrará en la tabla 1.4, las operaciones aritméticas:

Tabla 1. 4. Operación de matrices con operadores aritméticos en MATLAB

Comando	Descripción	Ejemplo
$A + B$	Suma las matrices A y B	$fx \gg A = [1\ 3\ 7; 4\ 7\ 1; 1\ 1\ 1]$ $A = \begin{matrix} 1 & 3 & 7 \\ 4 & 7 & 1 \\ 1 & 1 & 1 \end{matrix}$ $fx \gg B = [1\ 0\ 4; 1\ 2\ 2; 2\ 3\ 1]$ $B = \begin{matrix} 1 & 0 & 4 \\ 1 & 2 & 2 \\ 2 & 3 & 1 \end{matrix}$ $fx \gg A + B$ $ans = \begin{matrix} 2 & 3 & 11 \\ 5 & 9 & 3 \\ 3 & 4 & 2 \end{matrix}$
$A - B$	Resta las matrices A y B	$fx \gg A - B$ $ans = \begin{matrix} 0 & 3 & 3 \\ 3 & 5 & -1 \\ -1 & -2 & 0 \end{matrix}$
$A * B$	Multiplica las matrices A y B	$fx \gg A * B$ $ans = \begin{matrix} 18 & 27 & 17 \\ 13 & 17 & 31 \\ 4 & 5 & 7 \end{matrix}$
A/B	Divide las matrices A y B	$fx \gg A/B$ $ans = \begin{matrix} 0.25 & 3.75 & -1.50 \\ -0.5 & 0.50 & 2 \\ 0.25 & -0.25 & 0.50 \end{matrix}$
$A\hat{}n$	Potenciación de la matriz A	$fx \gg A\hat{}3$ $ans = \begin{matrix} 161 & 294 & 188 \\ 317 & 569 & 329 \\ 59 & 104 & 63 \end{matrix}$

1.5.6. OPERACIONES ESPECIFICAS CON MATRICES

A continuación, listaremos algunos comandos, que serán de utilidad en esta tesis. Tomaremos como ejemplo la matriz A de la tabla 1.4

Tabla 1. 5 Operaciones específicas de matrices en MATLAB

Comando	Descripción	Ejemplo
λA	Multiplica por un escalar la matriz A	$fx \gg 5 * A$ $ans = \begin{matrix} 5 & 15 & 35 \\ 20 & 35 & 5 \\ 5 & 5 & 5 \end{matrix}$
A'	Genera la transpuesta de la matriz A	$fx \gg A'$ $ans = \begin{matrix} 1 & 4 & 1 \\ 3 & 7 & 1 \\ 7 & 1 & 1 \end{matrix}$
$inv(A)$	Genera la inversa de la matriz A	$fx \gg inv(A)$ $ans = \begin{matrix} -0.25 & -0.1667 & 1.9167 \\ 0.1250 & 0.25 & -1.1250 \\ 0.1250 & -0.0833 & 0.2083 \end{matrix}$
$diag(A)$	Vector conteniendo la diagonal principal de la matriz A	$fx \gg diag(A)$ $ans = \begin{matrix} 1 \\ 7 \\ 1 \end{matrix}$
$det(A)$	Calcula el determinante de la matriz A	$fx \gg det(A)$ $ans = -24$
$ones(n)$	Matriz $n\ x\ n$ con todas sus componentes iguales a uno	$fx \gg ones(3)$ $ans = \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$
$eye(n)$	Devuelve la matriz Identidad de orden $n\ x\ n$	$fx \gg eye(3)$ $ans = \begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$

1.5.7. VALOR Y VECTOR PROPIO

MATLAB cuenta con varias funciones, que calcula los valores y vectores propios de una matriz mediante la solución de la ecuación $Av = \lambda v$. Tomaremos como ejemplo la matriz M, para describir los comandos listados en la tabla 1.6.

$$M = \begin{pmatrix} 0 & 0 & 1/2 & 1/2 & 1/3 \\ 0 & 0 & 0 & 0 & 1/3 \\ 1/3 & 0 & 0 & 1/2 & 1/3 \\ 1/3 & 0 & 1/2 & 0 & 0 \\ 1/3 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Tabla 1. 6. Operaciones específicas de matrices en MATLAB.

Comando	Descripción	Ejemplo
$poly$	Muestra el polinomio característico de la matriz $p(\lambda) = \lvert M - \lambda I \rvert$, cuyos coeficientes son los elementos del vector fila.	$fx \gg poly(M)$ $ans = 1.0 - 0.00 - 1.0278 \;\; 0.222 \;\; 0.1944 \;\; 0.0556$
$roots$	Calcula las raíces del polinomio característico $p(\lambda)$	$fx \gg roots(poly(M))$ $ans = \begin{matrix} 1.0000 \\ 0.5000 \\ 0.6667 \\ 0.5000 \\ 0.3333 \end{matrix}$
$[V, D] = eig$	Calcula una matriz D, con los valores propios de M en la diagonal principal, y a la matriz V de vectores propios, correspondientes a los valores propios de M.	$fx \gg [V, D] = eig(M)$ $V = \begin{matrix} 0.6117 & 0.2673 & -0.6325 & 0.0000 & 0.3207 \\ 0.1019 & 0.2673 & 0.3162 & -0.0000 & -0.4276 \\ 0.5437 & 0.5345 & -0.0000 & -0.7071 & 0.2673 \\ 0.4757 & -0.5345 & 0.6325 & -0.7071 & 0.4811 \\ 0.3058 & -0.5345 & -0.3162 & 0.0000 & -0.6414 \end{matrix}$ $D = \begin{matrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0.6667 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0.3333 \end{matrix}$

Comando	Descripción	Ejemplo
eig	Calcula las raíces del polinomio característico $p(\lambda)$, sin necesidad de aplicar los comandos $poly$ y $roots$.	$fx \gg eig(M))$ $ans = \begin{matrix} 1.0000 \\ 0.5000 \\ 0.6667 \\ 0.5000 \\ 0.3333 \end{matrix}$
$norm$	La normalización de un vector consiste en obtener otro vector unitario de la misma dirección y sentido de un vector ya dado. $v = \frac{v}{\lVert v \rVert} = \frac{1}{\lVert v \rVert} v$ El cálculo manual seria: Tomamos un vector columna $v = \begin{pmatrix} 3 \\ 1 \\ 2 \\ 2 \end{pmatrix}$, lo normalizaremos, de tal manera que: $\frac{1}{\lVert v \rVert} v = \frac{1}{\sqrt{3^2 + 1^2 + 2^2 + 2^2}} \begin{pmatrix} 3 \\ 1 \\ 2 \\ 2 \end{pmatrix}$ $v = \begin{pmatrix} 0.7071 \\ 0.2357 \\ 0.4714 \\ 0.4714 \end{pmatrix}$	$\%\ Generaremos\ el\ vector\ columna(v)\%$ $fx \gg v = [3; 1; 2; 2]$ $v = \begin{matrix} 3 \\ 1 \\ 2 \\ 2 \end{matrix}$ $\%\ Normalizaremos\ el\ vector(v)\ \%$ $fx \gg v = v/norm(v)$ $v = \begin{matrix} 0.7071 \\ 0.2357 \\ 0.4714 \\ 0.4714 \end{matrix}$

1.5.8. PROGRAMACION EN MATLAB

La programación en MATLAB se realiza básicamente sobre archivos M, se le denomina de esta forma debido a su extensión m. Estos archivos son simples scripts y como tales, pueden ser creados y modificados desde cualquier editor de texto común; por ejemplo, el bloc de notas.

MATLAB incluye un editor de archivos M, orientado a la programación sobre este software. Si se opta por otro editor, se debe vigilar siempre que los archivos escritos se guarden con esta extensión.

De acuerdo a como se definan, estos archivos pueden separarse en dos tipos:

Archivos de comandos. Son archivos M que contienen instrucciones del MATLAB, una vez editados se los guarda con un nombre en algún directorio que se encuentre dentro del path de búsqueda del MATLAB. Por defecto, la carpeta de trabajo es el directorio bin, que se encuentra dentro de la carpeta de instalación del MATLAB.

Para citar los archivos M escritos, solo se escribir el nombre del archivo guardado en el prompt o línea de comandos del sistema $(fx \gg)$, y las instrucciones que hayan sido escritas serán ejecutadas.

Ejemplo: Se utilizará la resolución de una ecuación de segundo grado; inicialmente abrimos el editor de archivos M de MATLAB (clic en *new script*, del grupo *file*, de la ficha *home*), en el escribimos la siguiente línea, tal como muestra la figura 1.29.

Editor - D:\MATLAB\EcuacionSe.m

EcuacionSe.m

```
1 -    clear, clc
2 -    disp('Ecuacion de 2do Grado');
3 -    a=input('Ingresar el termino cuadratico:');
4 -    b=input('Ingresar el termino lineal:');
5 -    c=input('Ingresar el termino independiente:');
6      % Calcular la Discriminante%
7 -    D=b*b-4*a*c;
8 -    if D<0
9 -     fprintf('No tiene soluciones reales');
10 -    elseif D==0
11 -    SolucionUnica=-b/2*a;
12 -    fprintf('%.3f\n\n',SolucionUnica);
13 -    else
14 -    x1=(-b+sqrt(D))/2*a;
15 -    x2=(-b-sqrt(D))/2*a;
16 -    fprintf('%.3f\n\n',x1,x2);
17 -   end
```

Figura 1. 29. Programa para resolver una ecuación de segundo grado.

Guardamos el documento como *EcuacionSe*, ahora, ingresamos el nombre del archivo guardado en línea de comandos del sistema ($fx \gg$), sin la extensión m.

Supongamos que tenemos la ecuación $x^2 - 5x + 6$, en la línea de comandos del sistema queda:

$fx \gg EcuacionSe$

Ecuacion de 2do Grado

Ingresar el termino cuadratico:1

Ingresar el termino lineal:-5

Ingresar el termino independiente:6

3.000

2.000

Este es un ejemplo bastante básico, de que es lo que hace un archivo de comandos cuando es llamado. Aquí se debe notar algo importante, todas las variables que se hayan definido o creado dentro de este archivo, luego de su ejecución, pasaran a formar parte de nuestro espacio de trabajo o Workspace (Figura 1.30).

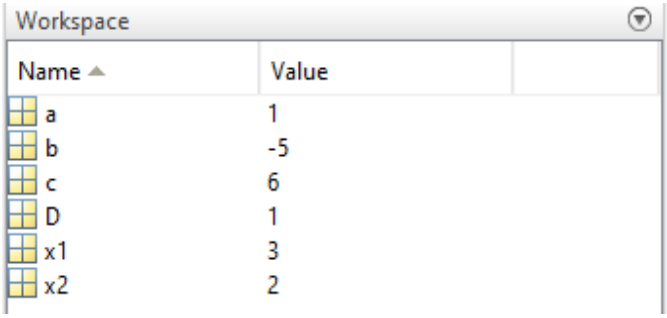


Figura 1. 30. Workspace con las variables del archivo de comando llamado.

Funciones. Una función es un archivo M similar al archivo M de comandos, con una diferencia en su definición. La primera línea de una función posee la siguiente estructura:

$$function[argumentos\_salida] = nombre(argumento\_entrada)$$

- *Nombre* correspondiente al nombre de la función.
- *Argumento_salida* representa una lista de elementos de salida, varia a medida que se ejecuta el programa. Los valores devueltos por la función citada serán los valores que se encuentran en *argumento_salida* en el momento en que termina la ejecución de la función. La definición de esta salida es opcional, de no aparecer, la función se convertiría en un procedimiento, puesto que no devuelve nada, solo acepta parámetros y se ejecuta el programa dado.
- *Arguementos_entrada* son los parámetros que recibe la función para poder realizar su procesamiento, estos argumentos durante la ejecución son parámetros recibidos por valor, es decir se hacen duplicados de los parámetros y sobre los cuales se realizan todas las modificaciones.

Contrario a los archivos M de comandos, todas las variables que se definen dentro del cuerpo o definición de la función, si la función es citada desde la línea de comando, no pasaran a formar parte del Workspace.

Es muy importante que el nombre de la función debe de coincidir con el nombre del archivo M con el cual se le guarda. De no hacer esto, podrían existir errores de directorio y/o ejecución.

Ejemplo. Vamos a crear una función que obtenga el conjunto solución para una ecuación de segundo grado, es decir obtendremos las raíces, de acuerdo a la condición de la discriminante, que será devuelto por la función.

Nuestra función ha de recibir tres parámetros, a, b y c, por lo tanto, estos se convertirán en los argumentos de la función, quedando como se observa en la figura 1.31.

Editor - D:\MATLAB\funCua.m

funCua.m

```
1      function [x1,x2]=funCua(a,b,c)
2
3 -    D=b*b-4*a*c;
4 -    if D<0
5 -     fprintf('Raices Imaginarias');
6 -    elseif D==0
7 -     x1=-b/2*a;
8 -    else
9 -     x1=(-b+sqrt(D))/2*a;
10 -    x2=(-b-sqrt(D))/2*a;
11
12 -   end
```

Figura 1. 31. Programa con función para resolver una ecuación de segundo grado.

Supongamos que tenemos la ecuación $x^2 - 5x + 6$, en la línea de comandos del sistema queda:

$$fx \gg [x1, x2] = funCua(1, -5,6)$$
$$x1 = 3$$
$$x2 = 6$$

Dentro del cuerpo de la función se puede salir de la función mediante el comando *return*, este comando detiene la ejecución del programa y devuelve el valor actual de las variables de salida a la función, procedimiento o rutina que lo cito.

1.6. BUSCADORES

1.6.1. DEFINICION

Los buscadores son los instrumentos más utilizados para localizar información en Internet. Un buscador es, en realidad, una herramienta que gestiona base de datos de Urls con distintos contenidos. Las definiciones aquí citadas se encuentran en [35] y [36].

La primera labor de un buscador consiste en enviar unos robots denominados arañas (Crawler), que exploran diversos grupos de páginas web, siguen los enlaces que contienen, exploran las páginas a las que estos enlaces se dirigen, y repiten su labor.

La información obtenida por el Crawler se almacena temporalmente en un repositorio central, el módulo de indización extrae una serie de datos, los comprime y almacena en varios índices.

Según Jesús Lamarca, la arquitectura de un buscador se basa en los siguientes cinco elementos fundamentales:

- **Robot (Crawler)**: Las bases de datos de los buscadores se suelen construir utilizando Crawler, es decir, programas que recorren la Web y recuperan los documentos de forma automática.
- **Indexador**: Programa que recibe las paginas recuperadas por el Crawler (normalmente el Crawler y el indexador son el mismo programa), extrae una representación interna de la misma y la vuelca en forma de índice en una base de datos.
- **Base de datos**: Colección de datos (organizados y estructurados) obtenidos de la indexación de las páginas web, a los que se puede acceder mediante un motor de búsquedas para hacer consultas y acceder a la información.

- **Motor de búsqueda**: Programa que se encarga de analizar una consulta realizada por el usuario y de buscar en el índice (base de datos) los documentos relacionados**.** Además, se encargan de ordenar los resultados de manera que aparezcan antes las páginas más relevantes.
- **Interfaz**: La interfaz más utilizada es la basada en páginas con formularios, donde el usuario introduce las palabras a buscar.

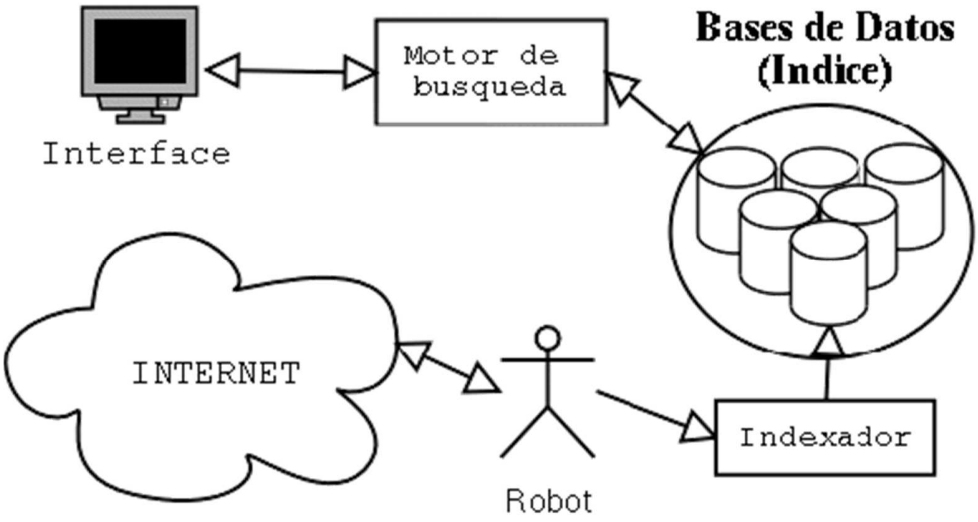


Figura 1. 32 . Esquema básico de la arquitectura de un buscador
Fuente: [35].

1.7. GOOGLE

Google es una empresa multinacional estadounidense especializada en servicios y productos relacionados con Internet, software, dispositivos electrónicos y otras tecnologías. Google aloja y desarrolla una serie de productos y servicios basados en Internet y la mayor parte de sus ingresos se genera a través de la publicidad de AdWords [37].

El término Google tiene origen en la matemática, google viene de *googol*, que es el número 10^{100}, es decir, el dígito 1 seguido de cien ceros. El *googol* tiene poca utilidad, sólo para explicar la diferencia entre un gran número y el infinito, y debido a su magnitud, los fundadores de Google decidieron adaptar el término para dar nombre a su empresa [37].

1.7.1. BREVE HISTORIA DE GOOGLE [38].

Google fue en sus inicios un proyecto de investigación de Lawrence Page y Sergei Brin (figura 1.33), dos doctorandos de la Universidad de Stanford (EEUU). A Page se le ocurrió investigar las características matemáticas de la red mundial (*www o world wide web*). Para ello, quería analizar y comprender la estructura de los enlaces entre las distintas páginas web.

Figura 1. 33. Lawrence Page y Sergei Brin
Fuente: Google.Inc

Su tutor, el profesor de ciencias de la computación Ferry Winograd le había animado a escoger ese tema. Page se lanzó a investigar qué páginas web enlazan a otra página, pensando que el número y la naturaleza de esos enlaces eran una información muy valiosa respecto a la página enlazada.

Estaba pensando en el papel de las citas académicas, de gran importancia en el mundo universitario y muy particularmente en Estados Unidos. Page llamó a ese proyecto inicial "*BackRub*". Su amigo Sergei Brin, un doctorando judío como Page de origen ruso, se unió enseguida al proyecto. El rastreador de Page comenzó a recorrer Internet en marzo de 1996. Tenía como base la página de la universidad de Stanford del propio Page. Para cristalizar los datos que recogía el rastreador, acerca de los enlaces dirigidos a cada página, ambos desarrollaron el algoritmo de PageRank (lo mencionaremos con más detalle en el siguiente capítulo).

Al analizar los resultados de *BackRub* que consistían en una lista de enlaces hacia una página concreta, clasificados por importancia pensaron que un buscador que tuviera en cuenta estos criterios produciría mejores resultados de búsqueda que los buscadores existentes en ese momento que sólo analizaban factores internos de una página web, como por ejemplo el número de veces que una palabra clave se repetía. Así nació el primer embrión del nuevo buscador, llamado Rankdex.

Registraron el dominio Google el 15 de septiembre de 1997. La firma Google, Inc. nació el 4 de septiembre de 1998 en el garaje de un amigo en Menlo Park, California.

Andreas von Bechtolsheim emprendedor germano estadounidense que había co-fundado la empresa Sun Microsystems en 1982, aportó la “semilla” inicial de capital de Google. Su cheque de 100.000 dólares, entregado en agosto de 1998, sería la mejor inversión de su vida.

Posteriormente, en junio de 1999 invirtieron en la compañía dos firmas de capital riesgo, *Sequioa Capital* y *Kleiner Perkins Caufield & Byers*, quienes hicieron una aportación de capital de 25 millones de dólares.

En un primer momento, los dos fundadores Brin y Page se oponían a colocar publicidad en su buscador. Pronto cambiarían de parecer. Posteriormente implementaron Google AdWords, un servicio y un programa de la misma, que se utiliza para ofrecer publicidad patrocinada a potenciales anunciantes. Los anuncios patrocinados de Adwords aparecen en simultáneamente a los resultados de las búsquedas. A finales de 1998, Google ya había indexado 60 millones de páginas.

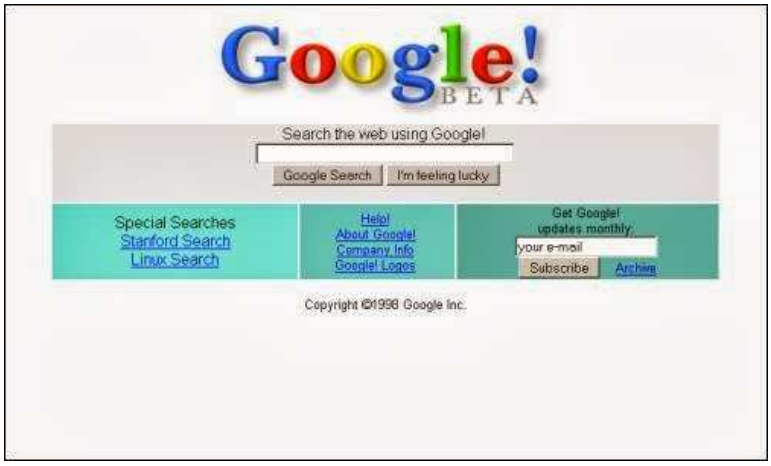


Figura 1. 34. Página principal de Google en septiembre de 1998
Fuente: Google.Inc.

La compañía se estableció en marzo de 1999 en Palo Alto, en el Valle del Silicio Silicon Valley, después de mudarse dos veces más por su rápido crecimiento.

Hoy en día Google atiende 200 millones de consultas diarias e indexa varios miles de millones de páginas web.

Una vez dominado el sector de las búsquedas, la exitosa compañía se ha expandido a una gran cantidad de sectores económicos que van más allá del negocio inicial. Ha lanzado su

propio servicio de correo electrónico Gmail, teléfonos móviles, su propio navegador para competir con Windows, Chrome y ha adquirido importantes compañías como YouTube, comprada por 1.650 millones de dólares el 9 de octubre de 2006 y DoubleClick, empresa de publicidad en Internet adquirida en abril de 2007 por 3.100 millones de dólares.

El verbo Google está incorporado al inglés. Desde 2006 está incluido en el diccionario de inglés de Oxford, el equivalente del diccionario de la Real Academia Española de la Lengua (RAE).

1.7.2. MEDICION DE LA IMPORTANCIA DE UN DOCUMENTO WEB [36] y [39].

Cada vez que efectuamos una búsqueda, Google nos informa de tres cosas: el número total de páginas encontradas, el tiempo que ha tardado en realizar esta búsqueda y una lista ordenada de las primeras páginas web encontradas por el buscador que tienen alguna relación con nuestra consulta.

El éxito de Google proviene del algoritmo de búsqueda y ordenación llamado PageRank, este método fue diseñado en 1998 en la universidad de Stanford por, Sergey Brin y Lawrence Page, cuando ambos eran estudiantes de doctorado de informática.

El PageRank, se basa en la naturaleza democrática de la Web, al utilizar la basta estructura de enlaces como un indicador del valor de una página individual. PageRank interpreta el enlace de una página A a una página B como un voto.

Pero el PageRank va más allá de contar el número de votos o enlaces que recibe una página; también analiza la página que emite el voto. Los votos emitidos por páginas que son importantes pesan más y ayudan a volver a otras páginas importantes. Ver figura 1.35.

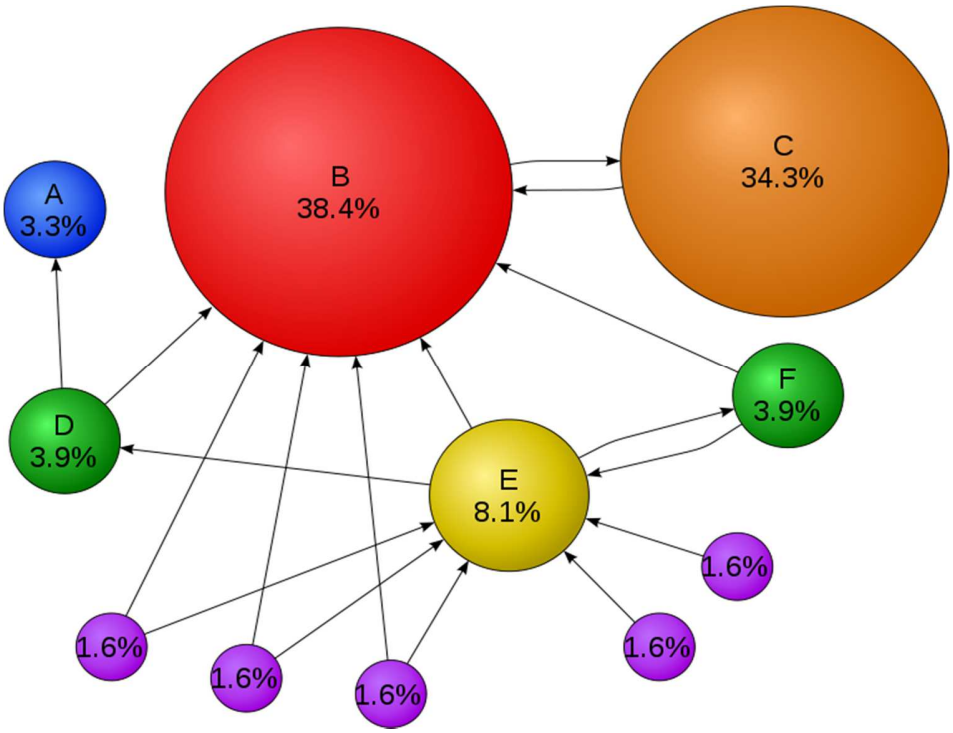


Figura 1. 35. Ejemplo grafico del PageRank
Fuente: https://en.wikipedia.org/wiki/File:PageRanks-Example.jpg

CAPÍTULO II

EL ALGOTIRMO DE ORDENACION DE GOOGLE

En la actualidad todos utilizamos el buscador de Google en muchas oportunidades, es más este hecho se ha convertido en algo natural y habitual para nosotros, pero rara vez nos paramos a pensar que hay detrás de esta sencilla página que al introducir una palabra y hacer un clic nos proporciona una lista de las páginas web pertinentes con nuestra consulta y que tal lista nos muestre ordenada según un criterio de importancia.

En esta capitulo vamos a modelizar de manera sencilla las páginas web que hay en la red y sus enlaces, con el fin de establecer un orden, de las páginas que se obtendrán como resultado de una búsqueda realizada por un usuario en Google.

Es conocido que la relevancia de una página en dicha búsqueda dependerá de muchos factores, como puedan ser el número de visitas, la confianza del sitio al que pertenece, la posición en la que aparecen los términos buscados, las referencias que se realicen a dicha página, es decir, una infinidad de objetos, que para esta tesis no será abordada.

Para simplificar el modelo, vamos a considerar que la importancia de una página web únicamente depende de las referencias que se hagan a ella; es decir, de los enlaces que apuntan a dicha página.

2.1. PLANTEAMIENTO DEL MODELO

Si hacemos una representación gráfica de la Web en la que suponemos que cada página es un punto y unimos dos de ellas, a y b, mediante una flecha si a enlaza con b, podemos visualizar la Web como un grafo dirigido de dimensiones colosales. Recordemos que un grafo viene determinado por dos conjuntos que denotamos por V cuyos elementos representan los vértices del grafo, y E, formado por pares de elementos de V, los cuales reciben el nombre de aristas del grafo.

En nuestro caso, no vamos a trabajar con la representación gráfica de este grafo, que sería imposible de documentarlo, si no que vamos a tomar una consideración más abstracta de este grafo, utilizando su matriz de adyacencia.

Definición 2 [19]. Dado un grafo $G = (V, E)$ con $V = \{v_1, v_2, \dots, v_n\}$, se denomina matriz de adyacencia a la matriz $M = (m_{ij})$ de orden $n\ x\ n$ que verifica

$$m_{ij} = 1, \quad si\ \{v_i, v_j\} \in E \tag{2.1}$$

$$m_{ij} = 0, \quad si\ \{v_i, v_j\} \notin E \tag{2.2}$$

Denotamos por P_i con $i = 1,2, \dots, n$, las distintas paginas web y por x_i la importancia de la pagina P_i. Teniendo en cuenta los enlaces entre páginas, podemos considerar la red como un grafo dirigido, cuyos vértices serán las páginas y las aristas serán los enlaces entre ellas. Es decir, habrá una arista de la pagina P_i a la P_j si en la pagina P_i hay un enlace a la pagina P_j [19] y [36].

$$\text{Conjuntos de páginas } P_i: \quad \{P_1, P_2, \dots, P_n\} \tag{2.3}$$

$$\text{importancia } x_i: \begin{pmatrix} x_1(P_1) \\ x_2(P_2) \\ \vdots \\ x_n(P_n) \end{pmatrix} \tag{2.4}$$

¿Cómo transformamos este grafico en un objeto apto para el cálculo? Ordenamos los vértices $P_1 \dots P_n$ y le asignamos a cada uno de ellos una fila y una columna en una matriz cuadrada [36]. Así la transpuesta de la matriz de adyacencia del grafo que hemos considerado que es con la que vamos a trabajar y que por simplicidad continuaremos denotando por M, tendrá un 1 en la entrada m_{ij} si hay un enlace de la pagina P_j a la pagina P_i y un 0 en caso contrario. Hemos construido la denominada matriz de adyacencia M del grafo dirigido [19] y [36]:

$$M = \begin{array}{c} \\ P_1 \\ \vdots \\ P_i \\ \vdots \\ P_n \end{array} \begin{array}{c} \begin{array}{ccccc} P_1 & \dots & P_j & \dots & P_n \end{array} \\ \begin{pmatrix} \vdots & \dots & \vdots & \dots & \vdots \\ \vdots & & \vdots & & \vdots \\ \vdots & \dots & m_{ij} & \dots & \vdots \\ \vdots & & \vdots & & \vdots \\ \vdots & \dots & \vdots & \dots & \vdots \end{pmatrix} \end{array} \quad m_{ij} = \begin{cases} 1 \; si \; hay \; enlace \; de \; P_j \; a \; P_i \\ 0 \; \; si \; no \; hay \; enlace \; de \; P_j \; a \; P_i \end{cases} \qquad \textbf{(2.5)}$$

De esta forma la fila i-ésima de la matriz estará determinada por los enlaces que hay a la pagina P_i desde los diferentes sitios P_j. Con la cual para saber el número de enlaces que hay a cada página, servirá con sumar las entradas de la fila correspondiente, De la misma manera, la columna i-ésima representa los enlaces que tiene la pagina P_i a los diferentes sitios de la red. Con lo cual tendremos que tener en cuenta que la matriz M que vamos a utilizar nos es simétrica [19].

Consideremos una pequeña web formado únicamente por cinco páginas P_i con $i = 1,2, \dots ,5$ que se relacionan como muestra la figura 2.1

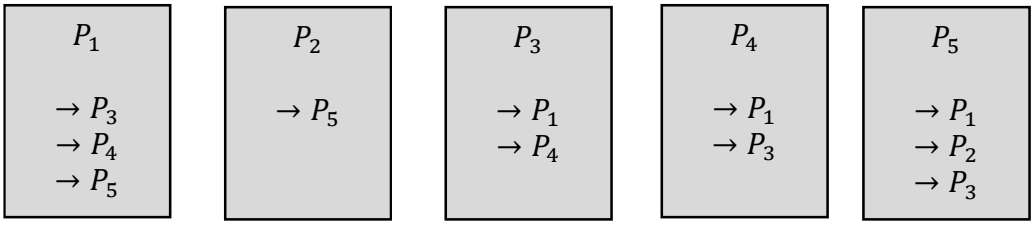


Figura 2. 1.Web de cinco páginas con sus respectivos enlaces

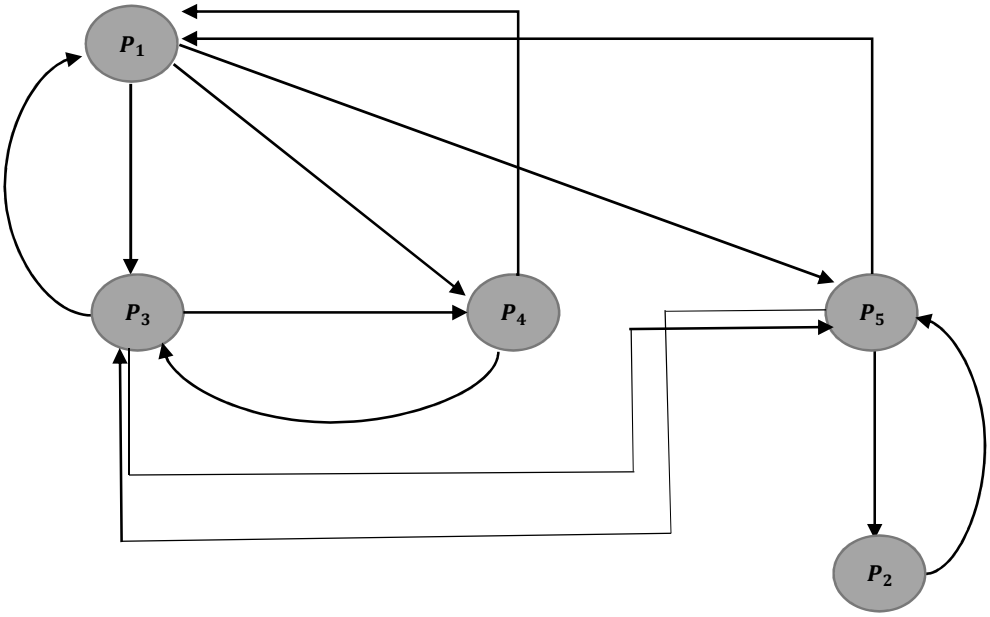


Figura 2. 2. Grafo dirigido de la figura 2.1

De esta manera la matriz asociada a esta Web sería la siguiente

$$M = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Como podemos observar la matriz M no es simétrica, pues nuestro grafo es dirigido.

En esta tesis no vamos a utilizar exactamente esta matriz, si no que vamos a utilizar una matriz M^*muy parecida y fácil de construir a partir de $M.$, únicamente es dividir cada término de la columna i por el número de unos que haya en dicha columna, es decir :

$$M^* = \begin{array}{c} \\ P_1 \\ \vdots \\ P_i \\ \vdots \\ P_n \end{array} \begin{array}{c} \begin{array}{ccccc} P_1 & \dots & P_j & \dots & P_n \end{array} \\ \begin{pmatrix} \vdots & \dots & \vdots & \dots & \vdots \\ \vdots & & \vdots & & \vdots \\ \vdots & \dots & \dfrac{m_{ij}}{n_j} & \dots & \vdots \\ \vdots & & \vdots & & \vdots \\ \vdots & \dots & \vdots & \dots & \vdots \end{pmatrix} \end{array} \quad m_{ij} = \begin{cases} 1 & si\ hay\ enlace\ de\ P_j\ a\ P_i \\ 0 & si\ no\ hay\ enlace\ de\ P_j\ a\ P_i \end{cases} \qquad \textbf{(2.6)}$$

Donde n_j es el número de enlaces en la pagina P_j ($n_j \neq 0$)

Entonces la matriz M^* para nuestro ejemplo quedaría de la siguiente forma.

$$M^* = \begin{pmatrix} 0 & 0 & 1/3 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 & 0 \\ 1/3 & 0 & 1/3 & 0 & 0 \\ 1/3 & 1 & 1/3 & 0 & 0 \end{pmatrix}$$

Observando la matriz M^* que acabamos de construir podemos ver que cada entrada es un número entre cero y uno y que cada columna suma uno; es decir nos encontramos ante una matriz estocástica o matriz de Markov. Esta matriz M^* es la que se utilizara como base de nuestro modelo.

2.2.EL ALGORITMO PAGERANK Y EL TEOREMA DE PERRON-FROBENIUS

Una vez que tenemos la web modelada, debemos decidir como vamos a asignar la importancia de cada página web para poder ordenarlos. Recordar que la importancia de la pagina P_i la denotamos por x_i.

Considerando que para este trabajo de tesis solo vamos a tener en cuenta los enlaces entre las paginas, la idea más intuitiva es que cuando más enlaces haya a una página más importante será. De esta forma obtendríamos la importancia de cada página sumando las entradas de la fila correspondiente de la matriz M. Bajo esta premisa las páginas de la figura 2.1 se ordenarán a partir del vector de importancias (3,1,3,2,2).

Como se puede percibir esta no es la situación que se pretende, ya que puede haber páginas que son referenciadas desde páginas muy importantes, pero que tengan pocas referencias. Para entender mejor este hecho imaginemos la siguiente situación:

Tenemos una página web con un artículo científico que tiene referencias desde *www.sopemat.org* (Sociedad Peruana de Educación Matemática) y *www.somape.org.pe* (Sociedad Matemática Peruana) y otra página distinta con otro artículo, enlazada desde www.miPaginaWeb.com y www.ejemplo.com, dos páginas ficticias supuestas de poca importancia. Con esta forma de medir la importancia de las páginas, ambas tendrían la misma valoración, pero es elemental que deberíamos darle más importancia a la primera debido a las páginas que la enlazan. Por este motivo debemos modificar la forma de asignar la importancia.

Así surge la idea, que consiste en que la importancia x_i de cada página P_i es directamente proporcional a la suma de las importancias de las páginas que la enlazan. De esta forma está claro que evitamos el problema que mostrábamos anteriormente.

En esta situación para la web del ejemplo 1 tendríamos que las importancias seguirán las siguientes relaciones [36]:

$$\begin{cases} x_1(P_1) = \frac{1}{\alpha}\big(x_3(P_3) + x_4(P_4) + x_5(P_5)\big) \\ x_2(P_2) = \frac{1}{\alpha}\big(x_5(P_5)\big) \\ x_3(P_3) = \frac{1}{\alpha}\big(x_1(P_1) + x_4(P_4)\big) \\ x_4(P_4) = \frac{1}{\alpha}\big(x_1(P_1) + x_3(P_3)\big) \\ x_5(P_5) = \frac{1}{\alpha}\big(x_1(P_1) + x_2(P_2) + x_3(P_3)\big) \end{cases} \tag{2.7}$$

Donde $\frac{1}{\alpha}$ es una cierta constante de proporcionalidad. Que representado en forma matricial tendria el siguiente aspecto.

$$\begin{pmatrix} x_1(P_1) \\ x_2(P_2) \\ x_3(P_3) \\ x_4(P_4) \\ x_5(P_5) \end{pmatrix} = \frac{1}{\alpha}\begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}\begin{pmatrix} x_1(P_1) \\ x_2(P_2) \\ x_3(P_3) \\ x_4(P_4) \\ x_5(P_5) \end{pmatrix}$$

Denotaremos por x el vector que contiene las importancias de las paginas web y llamamos λ a α^{-1} . Asi que podremos escribir que la asignacion de importancias que pretendemos es una solucion de :

$$M^* x = \lambda x \tag{2.8}$$

A la ecuacion le podemos expresar de una manera mas simple:

- sumando el inverso aditivo a λx por la derecha obteniendo el siguiente resultado.

$$M^* x - \lambda x = \lambda x - \lambda x \tag{2.9}$$

- Factorizamos el vector v , obteniendo.

$$(M^* - \lambda)x = 0 \tag{2.10}$$

- El elemento neutro en producto de matrices es la matriz identidad, entonces la ecuacion correspondiente seria.

$$(M^* - \lambda I)x = 0 \tag{2.11}$$

Donde I es una matriz identidad de orden $n\ x\ n$.

En esta situación para la web del ejemplo 1 asumiríamos que la forma matricial tendria el siguiente aspecto.

$$\left[\begin{pmatrix} 0 & 0 & 1/3 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 & 0 \\ 1/3 & 0 & 1/3 & 0 & 0 \\ 1/3 & 1 & 1/3 & 0 & 0 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}\right] \begin{pmatrix} x_1(P_1) \\ x_2(P_2) \\ x_3(P_3) \\ x_4(P_4) \\ x_5(P_5) \end{pmatrix} = 0$$

La cuestion se ha transformado en un problema de valores propios y vectores propios, de esta forma llegamos a que el vector de importancias buscado, es un vector propio de la matriz M^* que por supuesto debe de tener todas sus entradas positivas.

En estos momentos es cuando nos damos cuenta de que el teorema de Perron-Frobenius que hemos tratado anteriormente, puede tener una gran importancia, ya que bajo ciertas condiciones nos asegura la existencia del vector buscado.

Una vez visto que nos encontramos en una situacion en la que puede que nos sea util el teorema de Perron-Frobenius, es hora de comprobar que realmente podemos aplicarlo, es decir que estamos ante una matriz no negativa e irreductible.

Que la matriz M^* es no negativa está claro, debido a la forma de definirla, pero que la matriz M^* es irreducible es algo más complicado de comprobar. En el Ejemplo 1 sí que nos

encontrábamos en una situación ideal (es fácil ver que la matriz es irreducible), pero se pueden dar casos como los que vamos a ver en el siguiente caso en la que la matriz no es irreducible.

Caso 1. (Página no enlazada). Vamos a considerar una web formada por cuatro páginas de las cuales una de ellas no tenga enlaces salientes ni entrantes, es decir una web como la que se describe en la figura 2.2.

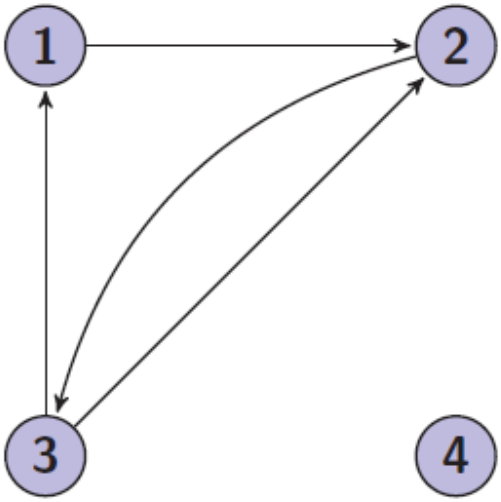


Figura 2. 3.Web de cuatro páginas con sus respectivos enlaces. Caso 1

Fuente: [19]

La matriz asociada a este grafo seria.

$$M^* = \begin{pmatrix} 0 & 0 & 1/2 & 0 \\ 1 & 0 & 1/2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Como podemos observar la matriz planteada es reducible, ya que cuenta con una columna, e incluso con fila de ceros.

Este tipo de páginas web aisladas (no tienen enlaces entrantes ni salientes) no la vamos a abordar en esta tesis.

Otro ejemplo que podemos encontrar es que haya un a página que no tenga enlaces salientes, algo habitual por otro lado, lo que haría que la matriz que estamos considerando no fuese irreducible. Veamos un caso que ilustre esta situación.

Caso 2. (Página web sin enlaces salientes). Tomamos una web formada por cuatro páginas estructuradas como en la figura 2.3

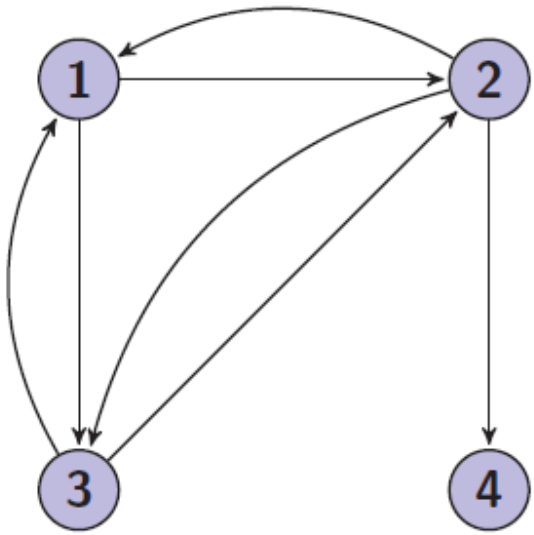


Figura 2. 4.Web de cuatro páginas con sus respectivos enlaces. Caso 2

Fuente: [19]

De esta forma la matriz asociada a este caso será.

$$M^* = \begin{pmatrix} 0 & 1/3 & 1/2 & 0 \\ 1/2 & 0 & 1/2 & 0 \\ 1/2 & 1/3 & 0 & 0 \\ 0 & 1/3 & 0 & 0 \end{pmatrix}$$

Observamos entonces que nos encontramos con una matriz que cuenta con una columna de ceros, por lo que es reducible.

En la situación descrita en el caso 2, podemos encontrarnos en una página que no tiene enlaces salientes, es decir no podemos desplazarnos a ninguna otra página. Cuando nos encontramos en una página como esta lo que hacemos es ir a cualquier otra página de la red. Esto podemos trasladarlo a nuestro modelo considerando que, si nos encontramos en una página sin enlaces salientes, podremos desplazarnos a cualquier otra página de la red con la misma probabilidad. Para representar este hecho en la matriz, sustituiremos la columna de ceros por una columna en la que cada elemento es $\frac{1}{n}$, donde n es el número de nodos de la red (paginas) [19]. De esta forma la matriz que resulta es la siguiente

$$M' = M^* + vd^t \tag{2.12}$$

Donde $v \in \mathbb{R}^n$ es el vector de probabilidad $v = \frac{1}{n}(1, \dots, 1)^t$ y $d^t \in \mathbb{R}^n$ se define tomando $d_i = 1$ si N_i=0 y $d_i = 0$ en otro caso, siendo i la columna con todas sus entradas nulas. El procedimiento y la forma matricial tendria el siguiente aspecto [19].

$$M' = \begin{pmatrix} 0 & 1/3 & 1/2 & 0 \\ 1/2 & 0 & 1/2 & 0 \\ 1/2 & 1/3 & 0 & 0 \\ 0 & 1/3 & 0 & 0 \end{pmatrix} + \frac{1}{4}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}^t$$

$$M' = \begin{pmatrix} 0 & 1/3 & 1/2 & 0 \\ 1/2 & 0 & 1/2 & 0 \\ 1/2 & 1/3 & 0 & 0 \\ 0 & 1/3 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 1/4 \\ 0 & 0 & 0 & 1/4 \\ 0 & 0 & 0 & 1/4 \\ 0 & 0 & 0 & 1/4 \end{pmatrix}$$

$$M' = \begin{pmatrix} 0 & 1/3 & 1/2 & 1/4 \\ 1/2 & 0 & 1/2 & 1/4 \\ 1/2 & 1/3 & 0 & 1/4 \\ 0 & 1/3 & 0 & 1/4 \end{pmatrix}$$

Nota: Si los elementos nulos se presentaran en las filas, M' sufrirá un ligero cambio. De esta forma la matriz que resulta es la siguiente:

$$M' = M^* + dv^t \tag{2.13}$$

Por ultimo presentamos el siguiente ejemplo que podemos encontrar al navegar por Internet y que por lo tanto debemos tener en cuenta.

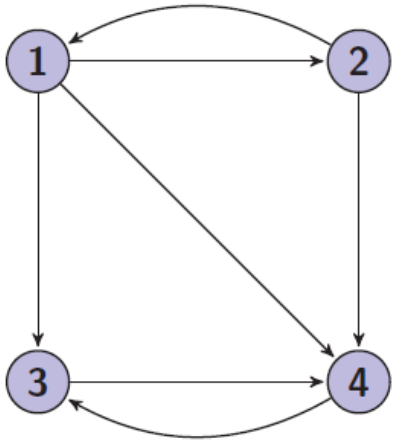


Figura 2. 5. Web de cuatro páginas con sus respectivos enlaces. Caso 3

Fuente: [19]

Caso 3 (Caso reducible). Tomamos ahora una web con la disposición establecida en la figura 2.4. La matriz que obtenemos en este caso es.

$$M' = \begin{pmatrix} 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1 \\ 1/3 & 1/2 & 1 & 0 \end{pmatrix}$$

Esta situación se puede dar de forma normal en la Web y se observa que esta matriz es reducible, con lo que no se puede aplicar el teorema de Perron-Frobenius.

Brin y Page se dieron cuenta de que en la Web aparecían este tipo de situación con lo que la matriz asociada no sería irreducible. Debido a esto modificaron un poco más el modelo creando una matriz M''de la forma [19].

$$M'' = \alpha M' + (1-\alpha)ve^t \tag{2.14}$$

Donde α es un valor comprendido entre 0 y 1[6], $e \in \mathbb{R}^n$ con $e^t = (1, \dots, 1)$ y v un vector de probabilidad que se suele tomar como en el caso anterior $v = \frac{1}{n}e$.

Así acabamos de conseguir una matriz M'' irreducible en todos los casos a costa de hacerla densa[7]. Este nuevo termino que hemos introducido $((1-\alpha)ve^T)$, se puede considerar como la probabilidad de que un usuario se canse de navegar siguiendo los enlaces y salte a una página cualquiera. Está claro que cuanto más cercano a uno sea el valor de α, más nos acercaremos a la matriz del grafo de la web, pero a costa de poder perder la irreductibilidad. El procedimiento y la forma matricial tendria el siguiente aspecto.

$$M'' = 0.85\begin{pmatrix} 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1 \\ 1/3 & 1/2 & 1 & 0 \end{pmatrix} + (1-0.85)\left(\frac{1}{4}\right)\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}\left(\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}^t\right)$$

$$M'' = 0.85\begin{pmatrix} 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1 \\ 1/3 & 1/2 & 1 & 0 \end{pmatrix} + (0.15)\begin{pmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{pmatrix}\left(\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}^t\right)$$

[6] Al parámetro α se le denomina damping (amortiguación) y se suele tomar $\alpha = 0.85$, que fue el valor tomado por Brin y Page. Este parámetro es objeto de estudio constante con el fin de mejorar el rendimiento del algoritmo [19].

[7] A esta matriz se le denomina matriz de Google y podemos ver que es una matriz irreducible y estocástica lo que es fundamental al calcular el vector de importancias [19].

$$M'' = 0.85\begin{pmatrix} 0 & 1/2 & 0 & 0 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1 \\ 1/3 & 1/2 & 1 & 0 \end{pmatrix} + (0.15)\begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix}$$

$$M'' = \begin{pmatrix} 0 & 17/40 & 0 & 0 \\ 17/60 & 0 & 0 & 0 \\ 17/60 & 0 & 0 & 17/20 \\ 17/60 & 17/40 & 17/20 & 0 \end{pmatrix} + \begin{pmatrix} 3/80 & 3/80 & 3/80 & 3/80 \\ 3/80 & 3/80 & 3/80 & 3/80 \\ 3/80 & 3/80 & 3/80 & 3/80 \\ 3/80 & 3/80 & 3/80 & 3/80 \end{pmatrix}$$

$$M'' = \begin{pmatrix} 3/80 & 37/80 & 3/80 & 3/80 \\ 77/240 & 3/80 & 3/80 & 3/80 \\ 77/240 & 3/80 & 3/80 & 71/80 \\ 77/240 & 37/80 & 71/80 & 3/80 \end{pmatrix}$$

Al construir la matriz de esta forma acabamos de conseguir que la matriz siempre sea irreducible, con lo que el teorema de Perron-Frobenius nos asegura la existencia del vector de ordenación buscado. Pero como la demostración de este Teorema no es constructiva, está claro que no nos ayudara a encontrarlo, debido a esto deberíamos recurrir a otros métodos para encontrar este vector. Además, es elemental que no podremos calcular todos los vectores propios asociados a la matriz y quedarnos con el asociado al del valor propio máximo, ya que la matriz es de proporcione gigantescas.

2.3.CALCULO DEL VECTOR PAGERANK

Hasta ahora hemos estado viendo como el teorema de Perron-Frobenius que hemos referenciando en la sección 1.1.6, nos asegura la existencia del vector PageRank, que da solución al problema de ordenación de las páginas web planteado por Brin y Page, pero nada hemos visto de como calcular este vector; es decir, sabemos que existe, pero desconocemos un algoritmo para calcularlo de forma eficiente.

Al explicar el método de ordenación en este capítulo hemos visto que el vector buscado es el vector propio correspondiente al valor propio máximo, pero está claro que no podemos calcular todos los vectores propios y valores propios y quedarnos con el que nos interesa, ya que las proporciones de la matriz de Google son desorbitantes, lo que hace imposible su cálculo.

Así para el cálculo de este vector se suele utilizar el método de la potencia que hemos referenciando en la sección 1.1.7, y será referenciado nuevamente de manera resumida.

Para este método es importante el hecho de que el vector buscado es el asociado al valor propio máximo. Supongamos que M es diagonalizable, entonces los vectores propios de la matriz M forman una base de $\mathbb{R}^n$, Ordenamos ahora los vectores propios de M de la forma $\{v_1, \dots v_n\}$ tal que los valores propios correspondientes estén ordenados como sigue [19].

$$|\boldsymbol{\lambda_1}| > |\boldsymbol{\lambda_2}| \geq |\boldsymbol{\lambda_3}| \geq \cdots |\boldsymbol{\lambda_n}| \tag{2.15}$$

Tomamos un vector $v \geq 0$ de $\mathbb{R}^n$ y como los vectores propios de M forman una base $\mathbb{R}^n$ ponemos el vector v en función de estos vectores propios.

$$v = c_1 v_1 + c_2 v_2 + \cdots + c_n v_n \tag{2.16}$$

Donde los valores $c_1, \dots, c_n$, que no necesitaremos calcular explícitamente, sino que bastaría con saber que existen, son las coordenadas del vector v en la base tomada.

Multiplicamos a la izquierda por la matriz M el vector v y como los vectores $\{v_1, \dots v_n\}$ son vectores propios de la matriz M obtenemos.

$$Mv = \lambda_1 c_1 v_1 + \lambda_2 c_2 v_2 + \cdots + \lambda_n c_n v_n \tag{2.17}$$

Notar que solo necesitamos calcular la parte izquierda de la igualdad y saber que la parte derecha existe. Seguimos multiplicando el vector v por la matriz M hasta obtener.

$$M^k v = {\lambda_1}^k c_1 v_1 + {\lambda_2}^k c_2 v_2 + \cdots + {\lambda_n}^k c_n v_n \tag{2.18}$$

Suponiendo que $c_1 \neq 0$ dividimos a ambos lados por la $k - ésima$ potencia del valor propio de modulo máximo.

$$\frac{1}{{\lambda_1}^k} M^k v = c_1 v_1 + \left(\frac{\lambda_2}{\lambda_1}\right)^k c_2 v_2 + \cdots + \left(\frac{\lambda_n}{\lambda_1}\right)^k c_n v_n \tag{2.19}$$

Además, como λ_1 es el valor propio del módulo máximo $\left|\frac{\lambda_i}{\lambda_1}\right|^k \leq 1$ para todo $i = 2, \dots, n$ y de esta forma.

$$\frac{1}{{\lambda_1}^k} M^k v \xrightarrow{k \to \infty} c_1 v_1 \tag{2.20}$$

Así a medida que vamos multiplicando por la matriz M vamos consiguiendo de manera más exacta el vector buscado, con una velocidad que depende del cociente entre los dos primeros valores propios. Este método es conocido como el método de las potencias (descrito en la sección 1.1.7).

CAPÍTULO III

PLANTEAMIENTO DE LOS CASOS DE ESTUDIO

Hasta ahora hemos estado viendo el algoritmo del PageRank a nivel teórico, pero en esta sección vamos a ver casos prácticos, empleando todas las definiciones vistas en las secciones preliminares.

3.1.CASO I: CALCULO DEL VECTOR DE IMPORTANCIAS

Retomando el ejemplo de la Figura 2.1, en donde la web de cinco páginas con sus respectivos enlaces, se le traslada a un grafo dirigido como muestra la figura 2.2, para luego expresarlo como la matriz adyacente M, de acuerdo a sus enlaces correspondientes.

$$M = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Posteriormente se construyó la matriz M^* , a partir de la matriz M, dividiendo cada término de la columna i por el número de unos que haya en dicha columna.

$$M^* = \begin{pmatrix} 0 & 0 & 1/3 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 & 0 \\ 1/3 & 0 & 1/3 & 0 & 0 \\ 1/3 & 1 & 1/3 & 0 & 0 \end{pmatrix}$$

Observando la matriz M^* podemos ver que cada entrada es un numero entre cero y uno y que cada columna suma uno; es decir nos encontramos ante una matriz estocástica o matriz de Markov.

Como se mencionó en secciones preliminares, los enlaces procedentes de páginas muy importantes aportan más notoriedad que los procedentes de páginas desconocidas. Entonces, la importancia x_i de cada página P_i es directamente proporcional a la suma de las importancias de las páginas que la enlazan.

Entonces, tendríamos que las importancias seguirán las siguientes relaciones:

$$\begin{cases} x_1(P_1) = \frac{1}{\alpha}\big(x_3(P_3) + x_4(P_4) + x_5(P_5)\big) \\ x_2(P_2) = \frac{1}{\alpha}\big(x_5(P_5)\big) \\ x_3(P_3) = \frac{1}{\alpha}\big(x_1(P_1) + x_4(P_4)\big) \\ x_4(P_4) = \frac{1}{\alpha}\big(x_1(P_1) + x_3(P_3)\big) \\ x_5(P_5) = \frac{1}{\alpha}\big(x_1(P_1) + x_2(P_2) + x_3(P_3)\big) \end{cases}$$

Representado en forma matricial tendria el siguiente aspecto.

$$\begin{pmatrix} x_1(P_1) \\ x_2(P_2) \\ x_3(P_3) \\ x_4(P_4) \\ x_5(P_5) \end{pmatrix} = \frac{1}{\alpha}\begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}\begin{pmatrix} x_1(P_1) \\ x_2(P_2) \\ x_3(P_3) \\ x_4(P_4) \\ x_5(P_5) \end{pmatrix}$$

Expresaremos por x el vector que contiene las importancias de las paginas web y llamamos λ a α^{-1} . Entonces la asignacion de importancias que pretendemos es una solucion de :

$$M''x = \lambda x$$

Esta igualdad se convirtió en un problema de valores propios y vectores propios, en el cual el vector de importancia x_i, es el vector propio de la matriz M''(ecuación 2.14), que por supuesto debe de tener todas sus entrandas positivas. A esta igualdad lo podemos expresar.

$$(M'' - \lambda I)x = 0$$

Donde I es una matriz identidad de orden $n\ x\ n$ y M'' es:

$$M'' = 0.85\begin{pmatrix} 0 & 0 & 1/3 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 & 0 \\ 1/3 & 0 & 1/3 & 0 & 0 \\ 1/3 & 1 & 1/3 & 0 & 0 \end{pmatrix} + (1 - 0.85)\left(\frac{1}{5}\right)\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}\left(\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}^t\right)$$

$$M'' = \begin{pmatrix} 0.0300 & 0.0300 & 0.3133 & 0.4550 & 0.4550 \\ 0.0300 & 0.0300 & 0.0300 & 0.0300 & 0.4550 \\ 0.3133 & 0.0300 & 0.0300 & 0.4550 & 0.0300 \\ 0.3133 & 0.0300 & 0.3133 & 0.0300 & 0.0300 \\ 0.3133 & 0.8800 & 0.3133 & 0.0300 & 0.0300 \end{pmatrix}$$

Escrita en forma matricial, queda expresado.

$$\left[\begin{pmatrix} 0.0300 & 0.0300 & 0.3133 & 0.4550 & 0.4550 \\ 0.0300 & 0.0300 & 0.0300 & 0.0300 & 0.4550 \\ 0.3133 & 0.0300 & 0.0300 & 0.4550 & 0.0300 \\ 0.3133 & 0.0300 & 0.3133 & 0.0300 & 0.0300 \\ 0.3133 & 0.8800 & 0.3133 & 0.0300 & 0.0300 \end{pmatrix} - \lambda\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}\right]\begin{pmatrix} x_1(P_1) \\ x_2(P_2) \\ x_3(P_3) \\ x_4(P_4) \\ x_5(P_5) \end{pmatrix} = 0$$

En adelante, aplicaremos algebra lineal, concretamente valores y vectores propio , para encontrar el valor propio máximo (λ_i) y el vector de importancias (x_i) . A continuacion detallamos los procedimientos:

Multiplicamos el escalar λ a la matriz Identidad (I) obteniendo:

$$\left[\begin{pmatrix} 0.0300 & 0.0300 & 0.3133 & 0.4550 & 0.4550 \\ 0.0300 & 0.0300 & 0.0300 & 0.0300 & 0.4550 \\ 0.3133 & 0.0300 & 0.0300 & 0.4550 & 0.0300 \\ 0.3133 & 0.0300 & 0.3133 & 0.0300 & 0.0300 \\ 0.3133 & 0.8800 & 0.3133 & 0.0300 & 0.0300 \end{pmatrix} - \begin{pmatrix} \lambda & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 \\ 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & 0 & \lambda \end{pmatrix}\right]\begin{pmatrix} x_1(P_1) \\ x_2(P_2) \\ x_3(P_3) \\ x_4(P_4) \\ x_5(P_5) \end{pmatrix} = 0$$

Se restará $(M'' - \lambda I)$ obteniendo.

$$\left[\begin{pmatrix} 0.0300-\lambda & 0.0300 & 0.3133 & 0.4550 & 0.4550 \\ 0.0300 & 0.0300-\lambda & 0.0300 & 0.0300 & 0.4550 \\ 0.3133 & 0.0300 & 0.0300-\lambda & 0.4550 & 0.0300 \\ 0.3133 & 0.0300 & 0.3133 & 0.0300-\lambda & 0.0300 \\ 0.3133 & 0.8800 & 0.3133 & 0.0300 & 0.0300-\lambda \end{pmatrix}\right]\begin{pmatrix} x_1(P_1) \\ x_2(P_2) \\ x_3(P_3) \\ x_4(P_4) \\ x_5(P_5) \end{pmatrix} = 0$$

Ahora tenemos que determinar los valores propios (λ), para ello calcularemos la determinante de $|(M'' - \lambda I)| = 0$

$$\left|\begin{pmatrix} 0.0300-\lambda & 0.0300 & 0.3133 & 0.4550 & 0.4550 \\ 0.0300 & 0.0300-\lambda & 0.0300 & 0.0300 & 0.4550 \\ 0.3133 & 0.0300 & 0.0300-\lambda & 0.4550 & 0.0300 \\ 0.3133 & 0.0300 & 0.3133 & 0.0300-\lambda & 0.0300 \\ 0.3133 & 0.8800 & 0.3133 & 0.0300 & 0.0300-\lambda \end{pmatrix}\right| = 0$$

Obteniendo como resultado el siguiente polinomio característico.

$$\lambda^5 - 0.15008\lambda^4 - 0.9303\lambda^3 - 0.0903\lambda^2 + 0.1416\lambda + 0.0290 = 0$$

De este polinomio tendremos las raíces.

$$\lambda_1 = 1, \quad \lambda_2 = 0.4555, \quad \lambda_3 = -0.7020, \quad \lambda_4 = -0.2833, \quad \lambda_5 = -0.3201$$

De estas raíces, tomaremos el valor cuyo valor absoluto sea el mayor $\lambda = max|\lambda_1|$

Entonces el valor propio dominante es $\lambda_1 = 1$, procedemos a reemplazarlo y restarlo en $(M'' - \lambda I)x = 0$ quedando expresado.

$$\left[\begin{pmatrix} -0.9700 & 0.0300 & 0.3133 & 0.4550 & 0.4550 \\ 0.0300 & -0.9700 & 0.0300 & 0.0300 & 0.4550 \\ 0.3133 & 0.0300 & -0.9700 & 0.4550 & 0.0300 \\ 0.3133 & 0.0300 & 0.3133 & -0.9700 & 0.0300 \\ 0.3133 & 0.8800 & 0.3133 & 0.0300 & -0.9700 \end{pmatrix}\right]\begin{pmatrix} x_1(P_1) \\ x_2(P_2) \\ x_3(P_3) \\ x_4(P_4) \\ x_5(P_5) \end{pmatrix} = 0$$

Multiplicaremos ambas matrices, obteniendo.

$$\begin{pmatrix} -0.9700x_1(P_1) & 0.0300x_2(P_2) & 0.3133x_3(P_3) & 0.4550x_4(P_4) & 0.4550x_5(P_5) \\ 0.0300x_1(P_1) & -0.9700x_2(P_2) & 0.0300x_3(P_3) & 0.0300x_4(P_4) & 0.4550x_5(P_5) \\ 0.3133x_1(P_1) & 0.0300x_2(P_2) & -0.9700x_3(P_3) & 0.4550x_4(P_4) & 0.0300x_5(P_5) \\ 0.3133x_1(P_1) & 0.0300x_2(P_2) & 0.3133x_3(P_3) & -0.9700x_4(P_4) & 0.0300x_5(P_5) \\ 0.3133x_1(P_1) & 0.8800x_2(P_2) & 0.3133x_3(P_3) & 0.0300x_4(P_4) & -0.9700x_5(P_5) \end{pmatrix} = 0$$

Igualando, obtenemos el siguiente sistema de ecuaciones.

$$\begin{array}{cccccc} -0.9700x_1(P_1) & 0.0300x_2(P_2) & 0.3133x_3(P_3) & 0.4550x_4(P_4) & 0.4550x_5(P_5) & = 0 \\ 0.0300x_1(P_1) & -0.9700x_2(P_2) & 0.0300x_3(P_3) & 0.0300x_4(P_4) & 0.4550x_5(P_5) & = 0 \\ 0.3133x_1(P_1) & 0.0300x_2(P_2) & -0.9700x_3(P_3) & 0.4550x_4(P_4) & 0.0300x_5(P_5) & = 0 \\ 0.3133x_1(P_1) & 0.0300x_2(P_2) & 0.3133x_3(P_3) & -0.9700x_4(P_4) & 0.0300x_5(P_5) & = 0 \\ 0.3133x_1(P_1) & 0.8800x_2(P_2) & 0.3133x_3(P_3) & 0.0300x_4(P_4) & -0.9700x_5(P_5) & = 0 \end{array}$$

Al resolver el sistema de ecuaciones, obtenemos:

$x_1(P_1) = 0.5574$, $x_2(P_2) = 0.3172$, $x_3(P_3) = 0.3606$, $x_4(P_4) = 0.3247$, $x_5(P_5) = 0.5944$

$$v = \begin{pmatrix} x_1(P_1) \\ x_2(P_2) \\ x_3(P_3) \\ x_4(P_4) \\ x_5(P_5) \end{pmatrix} = \begin{pmatrix} 0.5574 \\ 0.3172 \\ 0.3606 \\ 0.3247 \\ 0.5944 \end{pmatrix}$$

Los elementos del vector, serán ordenados de mayor a menor. De esta manera obtenemos el vector propio buscado, la cual nos da una lista ordenadas de las cinco páginas indexadas, según su relevancia, en este caso es.

$$v = \begin{pmatrix} x_5(P_1) \\ x_1(P_2) \\ x_3(P_3) \\ x_4(P_4) \\ x_2(P_5) \end{pmatrix} = \begin{pmatrix} 0.5944 \\ 0.5574 \\ 0.3606 \\ 0.3247 \\ 0.3172 \end{pmatrix}$$

Nos damos cuenta de que el teorema de Perron-Frobenius que hemos tratado anteriormente, tiene una gran importancia, ya que bajo ciertas condiciones nos asegura la existencia del vector buscado.

Este teorema nos indica que, si la matriz es cuadrada con entradas no negativas e irreductible, entonces existe un valor propio simple $\lambda > 0$, tal que $Av = \lambda v$, donde el vector propio correspondiente es $v > 0$. Además, el valor propio λ tiene multiplicidad algebraica y geométrica uno.

Para este caso, nos encontramos ante una matriz cuadrada de orden $5\ x\ 5$, con todas las entradas positivas e irreductible, y que existe un valor propio mayor a cero, además todos los elementos del vector propio son mayores a cero. Y el valor propio máximo es $\lambda = 1$, lo cual indica que tiene multiplicidad algebraica y geométrica uno.

Al explicar el método de ordenación en este primer caso, vimos que el vector buscado es el vector propio correspondiente al valor propio máximo, pero está claro que no podemos calcular todos los vectores propios y valores propios y quedarnos con el que nos interesa, ya que las proporciones de la matriz de Google son desorbitantes, lo que hace casi imposible su cálculo.

Así para el cálculo de este vector utilizaremos el método de la potencia, método iterativo que calcula sucesivas aproximaciones a los vectores propios y valores propios de una matriz, sin determinar el polinomio característico de la matriz en mención. Se usa principalmente para calcular el vector propio del mayor valor propio en matrices grandes. En particular, Google lo emplea para calcular el PageRank de las páginas webs.

3.2. CASO II: CALCULO DEL PAGERANK CON EL METODO DE LAS POTENCIAS

Consideramos una pequeña red formado por cuatro páginas, que se relaciona como muestra la figura 3.1

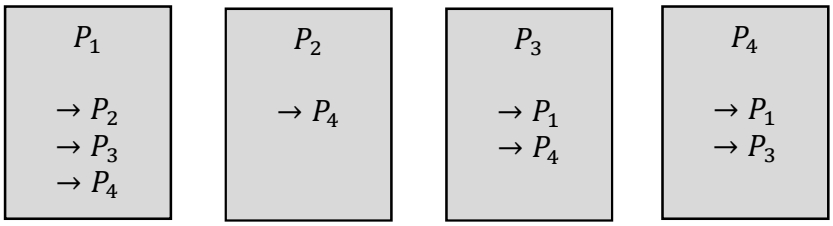


Figura 3. 1.Web de cinco páginas con sus respectivos enlaces. Caso práctico II

De esta manera la matriz asociada a esta red será la siguiente.

$$M = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Así la matriz de adyacencia y que por simplicidad continuaremos denotando por M, será representada por la ecuación 2.14:

$$M = 0.85\begin{pmatrix} 0 & 0 & 1/2 & 1/2 \\ 1 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 \\ 1/3 & 1/3 & 1/3 & 0 \end{pmatrix} + (1 - 0.85)\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}\left(\left(\frac{1}{5}\right)\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}^t\right)$$

$$M = \begin{pmatrix} 0.037500000000000 & 0.037500000000000 & 0.462500000000000 & 0.462500000000000 \\ 0.887500000000000 & 0.037500000000000 & 0.037500000000000 & 0.037500000000000 \\ 0.462500000000000 & 0.037500000000000 & 0.037500000000000 & 0.462500000000000 \\ 0.320833333333333 & 0.320833333333333 & 0.320833333333333 & 0.037500000000000 \end{pmatrix}$$

Tomaremos un vector x_0, que puede ser una aproximación inicial al vector de importancias, para tal fin calculamos la transpuesta $(M)^t$ de la matriz M y luego lo multiplicamos por la matriz M (norma de Frobenius) [40].

$$M^t = \begin{pmatrix} 0.037500000000000 & 0.887500000000000 & 0.462500000000000 & 0.320833333333333 \\ 0.037500000000000 & 0.037500000000000 & 0.037500000000000 & 0.320833333333333 \\ 0.462500000000000 & 0.037500000000000 & 0.037500000000000 & 0.320833333333333 \\ 0.462500000000000 & 0.037500000000000 & 0.462500000000000 & 0.037500000000000 \end{pmatrix}$$

$$M^tM = \begin{pmatrix} 1.105902777777778 & 0.154965277777778 & 0.170902777777778 & 0.276562500000000 \\ 0.154965277777778 & 0.107152777777778 & 0.123090277777778 & 0.048125000000000 \\ 0.170902777777778 & 0.123090277777778 & 0.31965277777777 & 0.244687500000000 \\ 0.276562500000000 & 0.048125000000000 & 0.244687500000000 & 0.430625000000000 \end{pmatrix}$$

Formaremos un vector de cuatro elementos con la diagonal principal de la matriz M^tM que lo llamaremos h_1 .

$$h_1 = (1.105902777777778 \quad 0.107152777777778 \quad 0.31965277777777 \quad 0.4306250000000006)^t$$

$$h_1 = \begin{pmatrix} 1.105902777777778 \\ 0.107152777777778 \\ 0.31965277777777 \\ 0.430625000000000 \end{pmatrix}$$

Normalizamos el vector inicial h_1, de esa manera obtenemos la aproximación inicial al vector dominante (x_0):

$$x_0 = \frac{h_1}{\|h_1\|} = \frac{1}{\|h_1\|} h_1$$

$$x_0 = \frac{\begin{pmatrix} 1.105902777777778 \\ 0.107152777777778 \\ 0.31965277777777 \\ 0.430625000000000 \end{pmatrix}}{\sqrt{(1.105902777777778)^2 + (0.107152777777778)^2 + (0.31965277777777)^2 + (0.430625000000000)^2}}$$

$$x_0 = \frac{1}{\sqrt{1.522118460648144}} \begin{pmatrix} 1.105902777777778 \\ 0.107152777777778 \\ 0.31965277777777 \\ 0.430625000000000 \end{pmatrix} = \begin{pmatrix} 0.896381164694376 \\ 0.086851876742444 \\ 0.259092150774770 \\ 0.349039849436096 \end{pmatrix}$$

Y finalmente tenemos la aproximación inicial al vector de importancias:

$$x_0 = \begin{pmatrix} 0.896381164694376 \\ 0.086851876742444 \\ 0.259092150774770 \\ 0.349039849436096 \end{pmatrix}$$

Como ya conocemos el vector x_0 que representa la aproximación inicial al vector de importancias, que será el punto de partida para los demás procesos iterativos.

Ahora encontraremos la siguiente aproximación del vector dominante que será x_1 para ello efectuamos la siguiente ecuación**:**

$$x_1 = \frac{(M^t M)x_0}{\|(M^t M)x_0\|}$$

Notamos la presencia del vector x_0 en la ecuación, deducimos entonces lo importante que es obtener esta aproximación inicial al vector de importancias.

Calculando $(M^t M)x_0$:

$$(\boldsymbol{M^t M})\boldsymbol{x_0} = \begin{pmatrix} 1.105902777777778 & 0.154965277777778 & 0.170902777777778 & 0.276562500000000 \\ 0.154965277777778 & 0.107152777777778 & 0.123090277777778 & 0.048125000000000 \\ 0.170902777777778 & 0.123090277777778 & 0.31965277777777 & 0.244687500000000 \\ 0.276562500000000 & 0.048125000000000 & 0.244687500000000 & 0.430625000000000 \end{pmatrix} \begin{pmatrix} 0.896381164694376 \\ 0.086851876742444 \\ 0.259092150774770 \\ 0.349039849436096 \end{pmatrix}$$

$$(M^t M)x_0 = \begin{pmatrix} 1.145580346815602 \\ 0.196903643592819 \\ 0.332109866482166 \\ 0.465787058235139 \end{pmatrix}$$

Normalizando $(M^tM)x_0$:

$$x_1 = \frac{\begin{pmatrix}1.1456\\0.1969\\0.3321\\0.4658\end{pmatrix}}{\sqrt{(1.145580346815602)^2 + (0.196903643592819)^2 + (0.332109866482166)^2 + (0.465787058235139)^2}}$$

$$x_1 = \frac{1}{\sqrt{1.678379922904430}}\begin{pmatrix}1.145580346815602\\0.196903643592819\\0.332109866482166\\0.465787058235139\end{pmatrix} = \begin{pmatrix}0.8843\\0.1520\\0.2564\\0.3595\end{pmatrix}$$

Finalmente, el valor de x_1:

$$x_1 = \begin{pmatrix}0.884260889020651\\0.151987760106727\\0.256351958729370\\0.359535915009561\end{pmatrix}$$

De forma similar calcularemos $x_2, x_{3,}x_4 \dots . x_i$, para aproximarnos al vector propio de importancias:

$$x_2 = \frac{(M^tM)x_1}{\|(M^tM)x_1\|}\ , x_3 = \frac{(M^tM)x_2}{\|(M^tM)x_2\|}\ \ , x_4 = \frac{(M^tM)x_3}{\|(M^tM)x_3\|} \dots\ x_i = \frac{(M^tM)x_{i-1}}{\|(M^tM)x_{i-1}\|}$$

Para una aproximación inicial al valor propio máximo (λ^1), multiplicamos el vector x_1 y su vector antecesor x_0:

$$\lambda^1 = {x_1}^t x_0 = \begin{pmatrix}0.884260889020651\\0.151987760106727\\0.256351958729370\\0.359535915009561\end{pmatrix}^t \begin{pmatrix}0.896381164694376\\0.086851876742444\\0.259092150774770\\0.349039849436096\end{pmatrix} = 0.997746369785489$$

De forma similar calcularemos $\lambda^{(1)}, \lambda^{(2)}, \lambda^{(3)} \dots \lambda^{(i)}$ para aproximarnos al valor propio máximo:

$$\lambda^{(2)} = {x_2}^t x_1,\ \lambda^{(3)} = {x_3}^t x_2,\ \lambda^{(4)} = {x_4}^t x_3 \ldots\ \lambda^{(i)} = {x_i}^t x_{i-1}$$

Continuando con la iteración x_2 y $\lambda^{(2)}$ (simplificaremos algunos procedimientos):

$$x_2 = \frac{(M^t M)x_1}{\|(M^t M)x_1\|}$$

$$x_2 = \frac{\begin{pmatrix} 1.105902777777778 & 0.154965277777778 & 0.170902777777778 & 0.276562500000000 \\ 0.154965277777778 & 0.107152777777778 & 0.123090277777778 & 0.048125000000000 \\ 0.170902777777778 & 0.123090277777778 & 0.31965277777777 & 0.244687500000000 \\ 0.276562500000000 & 0.048125000000000 & 0.244687500000000 & 0.430625000000000 \end{pmatrix} \begin{pmatrix} 0.884260889020651 \\ 0.151987760106727 \\ 0.256351958729370 \\ 0.359535915009561 \end{pmatrix}}{\left\| \begin{pmatrix} 1.105902777777778 & 0.154965277777778 & 0.170902777777778 & 0.276562500000000 \\ 0.154965277777778 & 0.107152777777778 & 0.123090277777778 & 0.048125000000000 \\ 0.170902777777778 & 0.123090277777778 & 0.31965277777777 & 0.244687500000000 \\ 0.276562500000000 & 0.048125000000000 & 0.244687500000000 & 0.430625000000000 \end{pmatrix} \begin{pmatrix} 0.884260889020651 \\ 0.151987760106727 \\ 0.256351958729370 \\ 0.359535915009561 \end{pmatrix} \right\|}$$

$$x_2 = \begin{pmatrix} 0.881323134245656 \\ 0.155655427591666 \\ 0.261576728918981 \\ 0.361411864488030 \end{pmatrix}$$

$$\lambda^{(2)} = {x_2}^t x_1 = \begin{pmatrix} 0.881323134245656 \\ 0.155655427591666 \\ 0.261576728918981 \\ 0.361411864488030 \end{pmatrix}^t \begin{pmatrix} 0.884260889020651 \\ 0.151987760106727 \\ 0.256351958729370 \\ 0.359535915009561 \end{pmatrix} = 0.999973550201061$$

Como observamos se trata de una secuencia de iteraciones, que en algún momento debería de parar, para ello se debe de considerar que el error relativo estimado sea menor al valor de tolerancia asignado(E) [40], El valor de tolerancia asignado por lo general es un número decimal que se aproxime a cero hasta en 16 casas decimales $(10^{-16} = 1.0e - 16)$

$$\left| \frac{\lambda^{(n)} - \lambda^{(n-1)}}{\lambda^{(n)}} \right| < E$$

Donde:

$$\lambda^{(k)} = Aproximación\ al\ autovalor\ en\ a\ iteracion\ n$$

$$\lambda^{(k-1)} = Aproximación\ al\ autovalor\ en\ a\ iteracion\ n - 1$$

$$E = Valor\ de\ tolerancia$$

Calculamos el error relativo estimado de $\lambda^{(2)}$ con un valor de tolerancia de $1.0e-16$.

$$\lambda^{(2)}: \left|\frac{\lambda^{(2)} - \lambda^{(1)}}{\lambda^{(2)}}\right| = \left|\frac{0.999973550201061 - 0.997746369785489}{0.999973550201061}\right| = 0.002227239325604$$

$$0.002227239325604 < E$$

$$0.002227239325604 < 1.0e-16 \ldots \text{No cumple}$$

Continuando con la iteración x_3 y $\lambda^{(3)}$:

$$x_3 = \frac{(M^t M)x_2}{\|(M^t M)x_2\|}$$

$$x_3 = \frac{\begin{pmatrix} 1.105902777777778 & 0.154965277777778 & 0.170902777777778 & 0.276562500000000 \\ 0.154965277777778 & 0.107152777777778 & 0.123090277777778 & 0.048125000000000 \\ 0.170902777777778 & 0.123090277777778 & 0.31965277777777 & 0.244687500000000 \\ 0.276562500000000 & 0.048125000000000 & 0.244687500000000 & 0.430625000000000 \end{pmatrix} \begin{pmatrix} 0.881323134245656 \\ 0.155655427591666 \\ 0.261576728918981 \\ 0.361411864488030 \end{pmatrix}}{\left\| \begin{pmatrix} 1.105902777777778 & 0.154965277777778 & 0.170902777777778 & 0.276562500000000 \\ 0.154965277777778 & 0.107152777777778 & 0.123090277777778 & 0.048125000000000 \\ 0.170902777777778 & 0.123090277777778 & 0.31965277777777 & 0.244687500000000 \\ 0.276562500000000 & 0.048125000000000 & 0.244687500000000 & 0.430625000000000 \end{pmatrix} \begin{pmatrix} 0.881323134245656 \\ 0.155655427591666 \\ 0.261576728918981 \\ 0.361411864488030 \end{pmatrix} \right\|}$$

$$x_3 = \begin{pmatrix} 0.880307400067806 \\ 0.156165254141953 \\ 0.263165378871184 \\ 0.362512452404363 \end{pmatrix}$$

$$\lambda^{(3)} = {x_3}^t x_2 = \begin{pmatrix} 0.880307400067806 \\ 0.156165254141953 \\ 0.263165378871184 \\ 0.362512452404363 \end{pmatrix}^t \begin{pmatrix} 0.881323134245656 \\ 0.155655427591666 \\ 0.261576728918981 \\ 0.361411864488030 \end{pmatrix} = 0.999997486629268$$

Calculamos error relativo estimado de $\lambda^{(3)}$:

$$\lambda^{(3)}: \left|\frac{\lambda^{(3)} - \lambda^{(2)}}{\lambda^{(3)}}\right| = \left|\frac{0.999997486629268 - 0.999973550201061}{0.999997486629268}\right| = 2.393648836857442e-5$$

$$2.393648836857442e-5 < 1.0e-16 \ldots \text{No cumple}$$

Continuando de esa manera obtenemos las siguientes iteraciones:

Iteración	$x_i = \frac{(M^tM)x_{i-1}}{\|(M^tM)x_{i-1}\|}$	$\lambda^{(i)} = {x_i}^t x_{i-1}$	$\left\|\frac{\lambda^{(i)} - \lambda^{(i-1)}}{\lambda^{(i)}}\right\| < E$	Cumple
4	$\begin{pmatrix}0.879942751902280\\0.156276740181535\\0.263677126442033\\0.362977556942778\end{pmatrix}$	0.999999688197385	$2.201568803709824e-6 < 1.0e-16$	No
5	$\begin{pmatrix}0.879811443973768\\0.156308070762270\\0.263853114541744\\0.363154435487151\end{pmatrix}$	0.999999959759396	$2.715620215841875e-7 < 1.0e-16$	No
6	$\begin{pmatrix}0.879764136001912\\0.156318208920931\\0.263915416781131\\0.363219403872925\end{pmatrix}$	0.999999994778357	$3.501896091407110e-8 < 1.0e-16$	No
7	$\begin{pmatrix}0.879747089005347\\0.156321710836101\\0.263937721266318\\0.363242978460197\end{pmatrix}$	0.999999999321943	$4.543585955002426e-9 < 1.0e-16$	No
8	$\begin{pmatrix}0.879740945915675\\0.156322952863745\\0.263945739775794\\0.363251495442982\end{pmatrix}$	0.999999999911942	$5.899997158018655e-10 < 1.0e-16$	No
9	$\begin{pmatrix}0.879738732131434\\0.156323397827133\\0.263948626878225\\0.363254567553743\end{pmatrix}$	0.999999999988564	$7.662170897847403e-11 < 1.0e-16$	No
10	$\begin{pmatrix}0.879737934343887\\0.156323557833824\\0.263949666978495\\0.363255675034084\end{pmatrix}$	0.999999999998515	$9.950706925125132e-12 < 1.0e-16$	No
11	$\begin{pmatrix}0.879737646842113\\0.156323615450464\\0.263950041759599\\0.363256074190513\end{pmatrix}$	0.999999999999807	$1.292632667571319e-12 < 1.0e-16$	No
12	$\begin{pmatrix}0.879737543233873\\0.156323636208011\\0.263950176815288\\0.363256218042738\end{pmatrix}$	0.999999999999975	$1.678657213233278e-13 < 1.0e-16$	No
13	$\begin{pmatrix}0.879737505896115\\0.156323643687708\\0.263950225485141\\0.363256269884258\end{pmatrix}$	0.999999999999997	$2.153832667772811e-14 < 1.0e-16$	No

Iteración	$x_i = \frac{(M^tM)x_{i-1}}{\|(M^tM)x_{i-1}\|}$	$\lambda^{(i)} = {x_i}^t x_{i-1}$	$\left\|\frac{\lambda^{(i)} - \lambda^{(i-1)}}{\lambda^{(i)}}\right\| < E$	Cumple
14	$\begin{pmatrix}0.879737492440540\\0.156323646383096\\0.263950243024414\\0.363256288566732\end{pmatrix}$	1	$2.775557561562893e-15 < 1.0e-16$	No
15	$\begin{pmatrix}0.879737487591494\\0.156323647354431\\0.263950249345107\\0.363256295299434\end{pmatrix}$	1	$2.220446049250314e-16 < 1.0e-16$	No
16	$\begin{pmatrix}0.879737485844021\\0.156323647704474\\0.263950251622922\\0.363256297725730\end{pmatrix}$	1	$2.220446049250313e-16 < 1.0e-16$	No
17	$\begin{pmatrix}0.879737485214277\\0.156323647830620\\0.263950252443788\\0.363256298600105\end{pmatrix}$	1	$4.440892098500624e-16 < 1.0e-16$	No
18	$\begin{pmatrix}0.879737484987333\\0.156323647876080\\0.263950252739607\\0.363256298915208\end{pmatrix}$	1	$2.220446049250313e-16 < 1.0e-16$	No
19	$\begin{pmatrix}0.879737484905548\\0.156323647892462\\0.263950252846213\\0.363256299028763\end{pmatrix}$	1	$2.220446049250313e-16 < 1.0e-16$	No
20	$\begin{pmatrix}0.879737484876075\\0.156323647898366\\0.263950252884631\\0.363256299069685\end{pmatrix}$	1	$0 < 1.0e-16$	Si

En la iteración veinte, nos detenemos porque el error relativo estimado es menor al valor de tolerancia ($1.0e-16$) por la tanto:

El valor propio máximo es:

$$\lambda = 1$$

Y su vector de importancias es:

$$v = \begin{pmatrix}0.879737484876075\\0.156323647898366\\0.263950252884631\\0.363256299069685\end{pmatrix}$$

Ordenamos de mayor a menor el vector de importancias:

$$v = \begin{pmatrix} P_1 \\ P_4 \\ P_3 \\ P_2 \end{pmatrix} \begin{pmatrix} 0.879737484876075 \\ 0.363256299069685 \\ 0.263950252884631 \\ 0.363256299069685 \end{pmatrix}$$

Las páginas serán listadas en el orden siguiente:

$$P_1, P_4, P_3, P_2$$

Para este caso, el teorema de Perron-Frobenius, cumple en todas las formas.

CAPÍTULO IV

SIMULACIÓN

A fin de entender las diferentes situaciones que el modelo puede enfrentar, una simulación será presentada en este capítulo. En particular estamos interesados en construir de manera aproximada el funcionamiento del motor de búsqueda Google. Para ello se presenta la ordenación según su relevancia de las páginas de un sitio web, según descrito en los capítulos anteriores.

4.1.INICIAR EL RASTREO DE LA PAGINAS

El Crawler empezará a rastrear, sobre el sitio web (dirección inicial (semillas)), y con los argumentos mostrados en la figura 4.1.

Figura 4. 1. Ventana de iniciación del rastreo

Al iniciar, comenzara indexando la página especificada y todas las páginas directamente enlazadas por la página de inicio.

En este proceso se indexo **6, 727** páginas en aproximadamente 96 horas y se encontró **2.421.966,00** relaciones (que página apunta a cuál) entre las páginas del sitio web. Como se hizo mención en sesiones anteriores, toda esta información es almacenada en las tablas ***ows_hosts*** y ***ows_index*** del gestor de base de datos Mysql.

4.2.CONSTRUIR EL ARCHIVO .NET DE PAJEK

A partir de las relaciones entre las paginas construiremos el archivo .net, para su posterior migrado al software *Pajek*, que nos permitirá visualizar el espacio web analizado por el Crawler y además este mismo, nos facilitará la integración con el entorno matemático *Matlab*. Para este fin se procederá:

- Abrimos un entorno grafico del *Mysql*, para este caso utilizaremos el *MySQL Workbench 6.3*, Observamos la presencia de las tablas ***ows_hosts*** y ***ows_index*** (Figura 4.2).

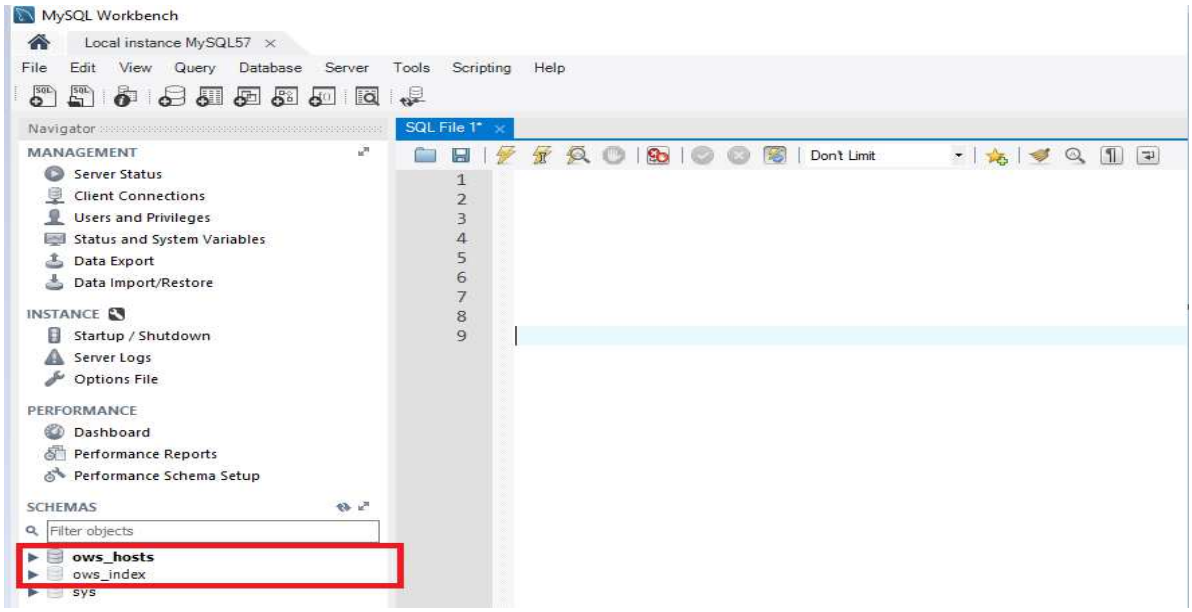


Figura 4. 2. Entorno grafico Mysql Workbench con las tablas *ows_host* y *ows_index*

- Mediante consultas SQL, generaremos el archivo .net, para ello digitamos el siguiente código en *MySQL Workbench 6.3*. Aclaramos que para este trabajo de tesis no consideraremos los enlaces internos de las paginas indexadas, solo tomaremos los enlaces externos. Por ese motivo en la línea de código 17, lo condicionamos con el operador < >. Ver figura 4.3.

```
1  select
2  '*Vertices' as Vertices,
3  count(*) as Total
4  from HostList
5  union all
6   select
7  id, concat('"',hostname,'"') as Vertices
8  from hostlist
9  Union all
10 select *from arcos
11 Union all
12 select
13 Distinct
14 host_id,linkedhost_id
15 from pages_map
16 where
17 host_id <>linkedhost_id
18 INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 5.7/Uploads/Tesis.net'
19 FIELDS TERMINATED BY ' '
20 LINES TERMINATED BY '\r\n'
```

Figura 4. 3. Consulta SQL para generar el archivo .net

- Al ejecutar la consulta, muestra un mensaje indicando la cantidad de filas afectadas y que la migración a un archivo .net en la dirección especificada, nombre y con el formato respectivo fue satisfactoria. Ver figura 4.4.

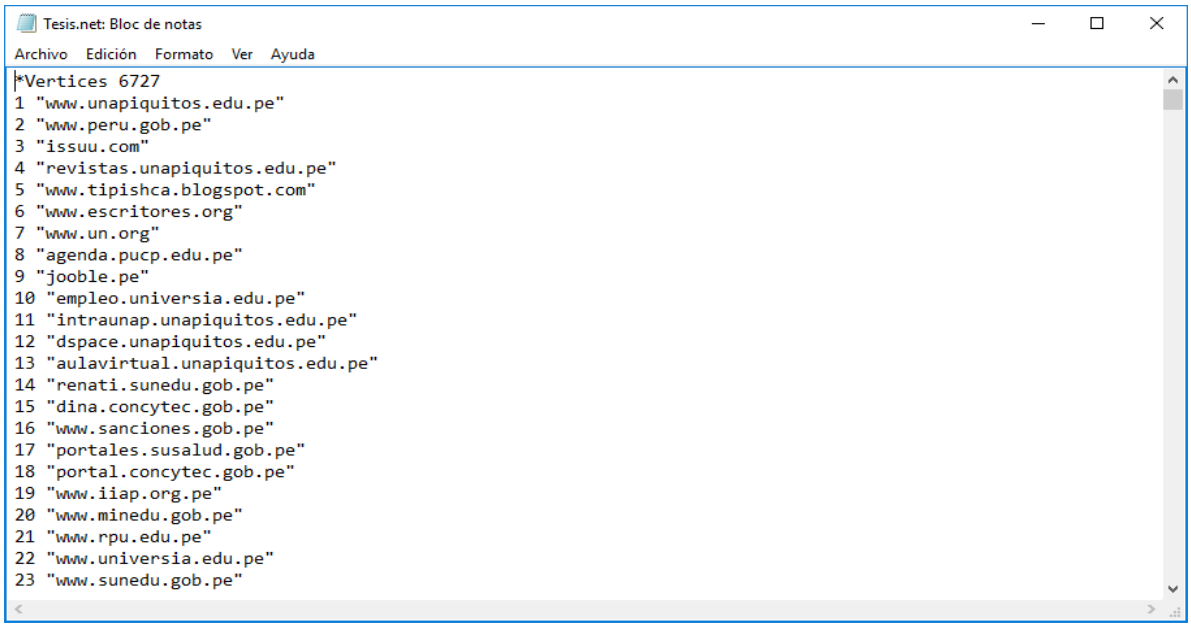
Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	19:25:09	select '*Vertices' as Vertices, count(*) as Total from HostList union all select id, concat('"',ho...	13689 row(s) affected	73.703 sec

Figura 4. 4. Mensaje indicando que la migración fue satisfactoria

- Al abrir el archivo *Tesis.net* nos mostrara los **Vertices* con sus actores(paginas) y **Arcs* que representa las relaciones (que página apunta a cuál). Ver figuras 4.5 y 4.6.

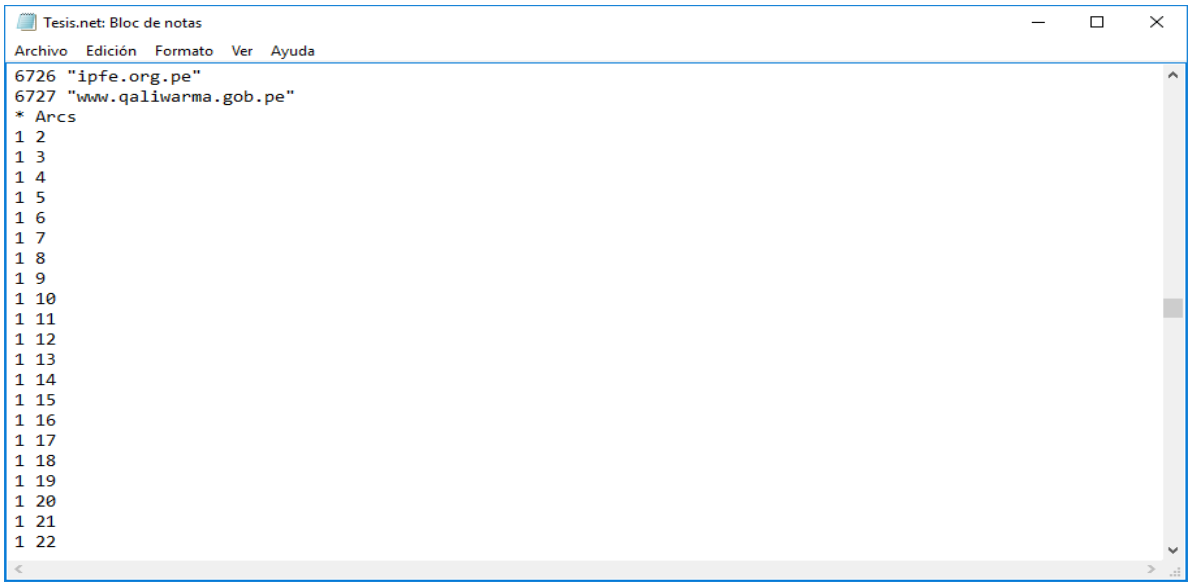


Tesis.net: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
*Vertices 6727
1 "www.unapiquitos.edu.pe"
2 "www.peru.gob.pe"
3 "issuu.com"
4 "revistas.unapiquitos.edu.pe"
5 "www.tipishca.blogspot.com"
6 "www.escritores.org"
7 "www.un.org"
8 "agenda.pucp.edu.pe"
9 "jooble.pe"
10 "empleo.universia.edu.pe"
11 "intraunap.unapiquitos.edu.pe"
12 "dspace.unapiquitos.edu.pe"
13 "aulavirtual.unapiquitos.edu.pe"
14 "renati.sunedu.gob.pe"
15 "dina.concytec.gob.pe"
16 "www.sanciones.gob.pe"
17 "portales.susalud.gob.pe"
18 "portal.concytec.gob.pe"
19 "www.iiap.org.pe"
20 "www.minedu.gob.pe"
21 "www.rpu.edu.pe"
22 "www.universia.edu.pe"
23 "www.sunedu.gob.pe"
```

Figura 4. 5. Lista de vértices del archivo .net



Tesis.net: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
6726 "ipfe.org.pe"
6727 "www.qaliwarma.gob.pe"
* Arcs
1 2
1 3
1 4
1 5
1 6
1 7
1 8
1 9
1 10
1 11
1 12
1 13
1 14
1 15
1 16
1 17
1 18
1 19
1 20
1 21
1 22
```

Figura 4. 6. Lista de los Arcs del archivo .net

4.3.CONSTRUIR EL GRAFO DE CONEXIONES EN PAJEK

Finalizado la construccion del archivo .net, construiremos el grafo o red de conexiones que nos permitara vizualizar el espacio web analizado por el Crawler, para tal fin procedemos:

- Abrimos el archivo *Tesis.net*, Ir al menú "*File*", posteriormente a "*Network*" y finalmente a "*Read*" para elegir el archivo mencionado y seleccionarlo. Automáticamente nos abrirá una ventana de "*Report*" donde se irán guardando el resultado de los procesos y si hubiera algún error lo indicara. Ver figura 4.7.

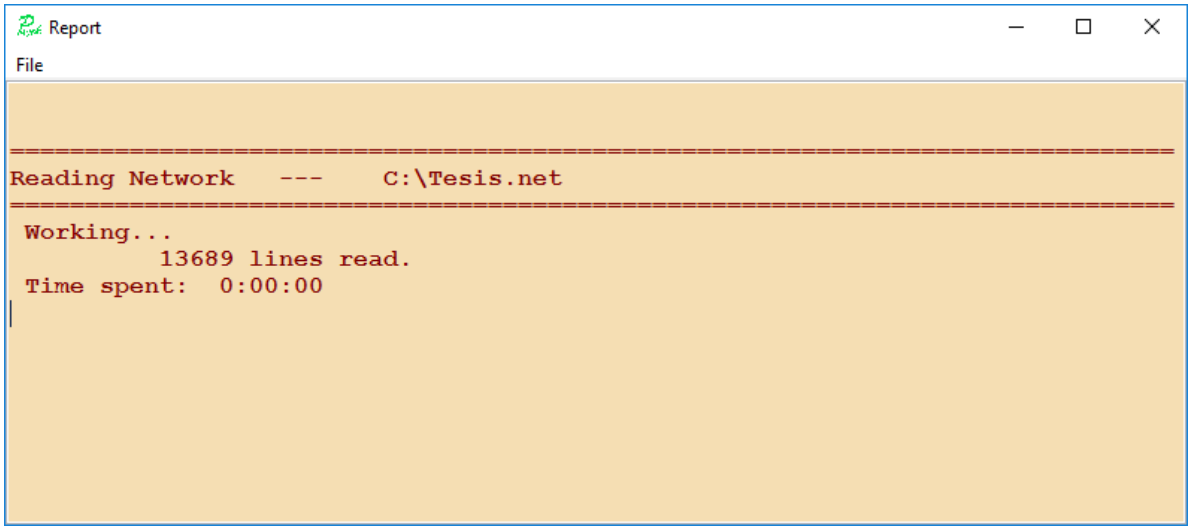


Figura 4. 7. Reporte del archivo Tesis.net

- Seguidamente, para observar la red que representa deberemos ir a la opción "*Draw*" y pulsar en la opción "*Network*". Así obtendremos nuestra red del archivo importado. Para su mejor visualización cambiaremos la red a la opción ***Layout Enegery/Fruchterman Reingold/3D***. Ver figura 4.8 y 4.9.

Figura 4. 8. Opciones ***Layout Enegery/Fruchterman Reingold/3D***

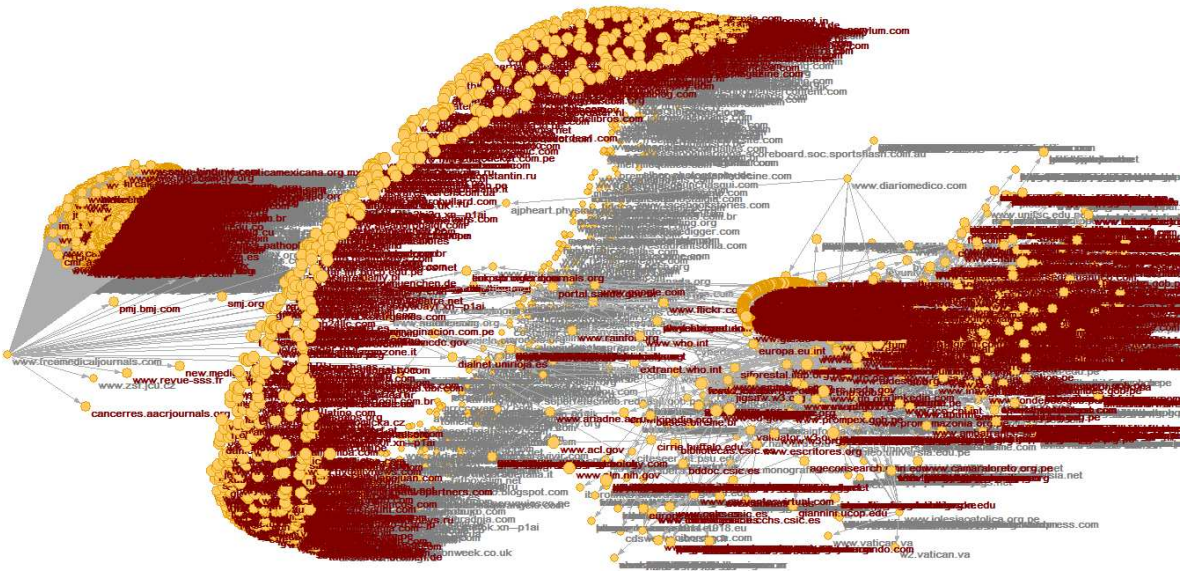


Figura 4. 9. Grafo del archivo Tesis.net en 3D

- Quitaremos las la etiquetas de los nodos(nombre de las paginas web), con el fin de observar el grafo de conexiones con mas profundidad, para tal fin, cambiamos a ***Options/ Mark Vertices Using /No Labels***. Ver figura 4.10.

- El nuevo grafo que se muestra en la figura 4.11, como observamos hay muchos nodos colgantes(vertices que no tienen enlaces ni entrantes ni salientes), la cual no seria optimo para el modelo matemematico a simular.

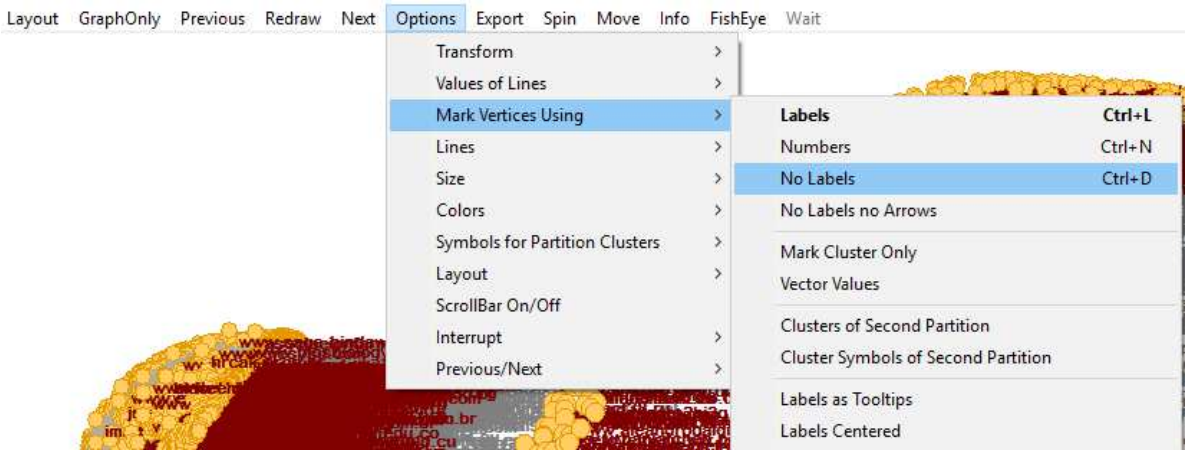


Figura 4. 10. Opciones ***Options/ Mark Vertices Using /No Labels***

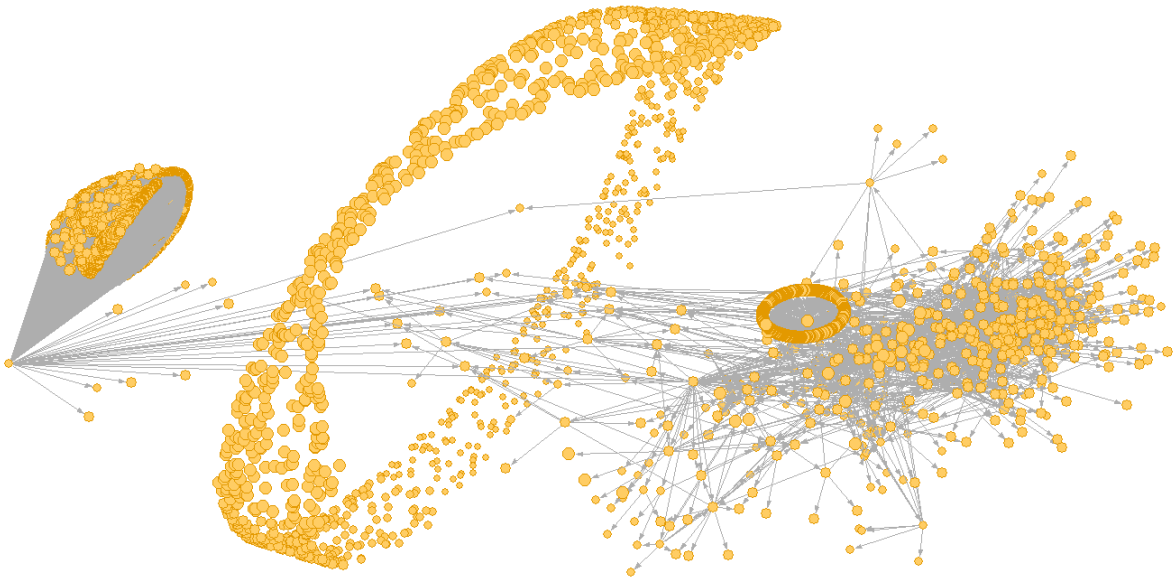


Figura 4. 11.Grafo del archivo Tesis.net sin etiquetas ni números

- Para reducir los vertices que no tienen vinculos, vamos a ***Network/Create New Network/ Transform/Reduction/Degree/All***. Ver figura 4.12

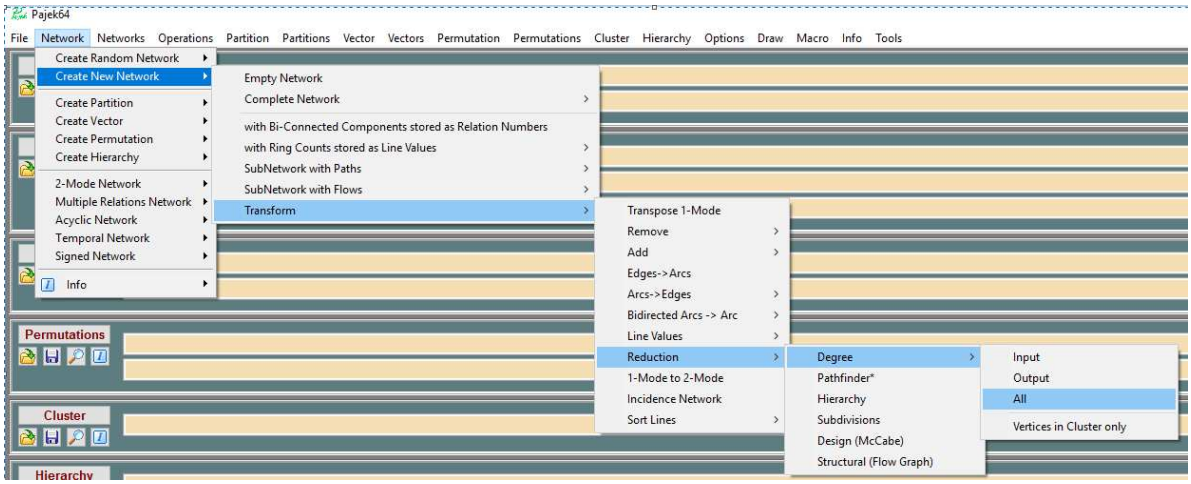


Figura 4. 12. Opciones ***Network/Create New Network/ Transform/Reduction/Degree/All***

- Seguidamente mostrara la ventana *Minimum degree of vertices*, que nos indica el grado que queremos para nuestra red, para este fin lo dejaremos con el valor de 2 , debido a que varios vertices tiene grados 0 (sin enlaces entrantes ni salientes), por lo tanto seran retirados de la red. Clic en *Ok* , automaticamente nos abrira una ventana "*Report*" donde se irán guardando el resultado de los procesos. Ver figura 4.14

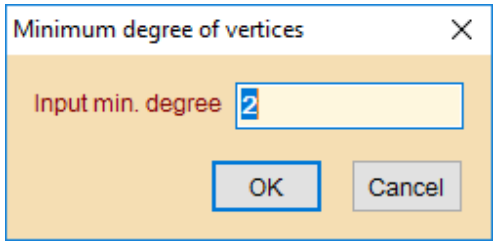


Figura 4. 13. Ventana Minimum degree of vertices

- Finalizado el proceso de reduccion, para observar la nueva red, debemos ir a la opcion "*Draw*" y pulsar en la opción "*Network*. Para una mejor visualización el nombre de cada página(vértice) será reemplazado por el número de identificación de la misma. Para tal fin nos vamos a ***Options/ Mark Vertices Using /Numbers***. Ver figura 4.15 y 4.16

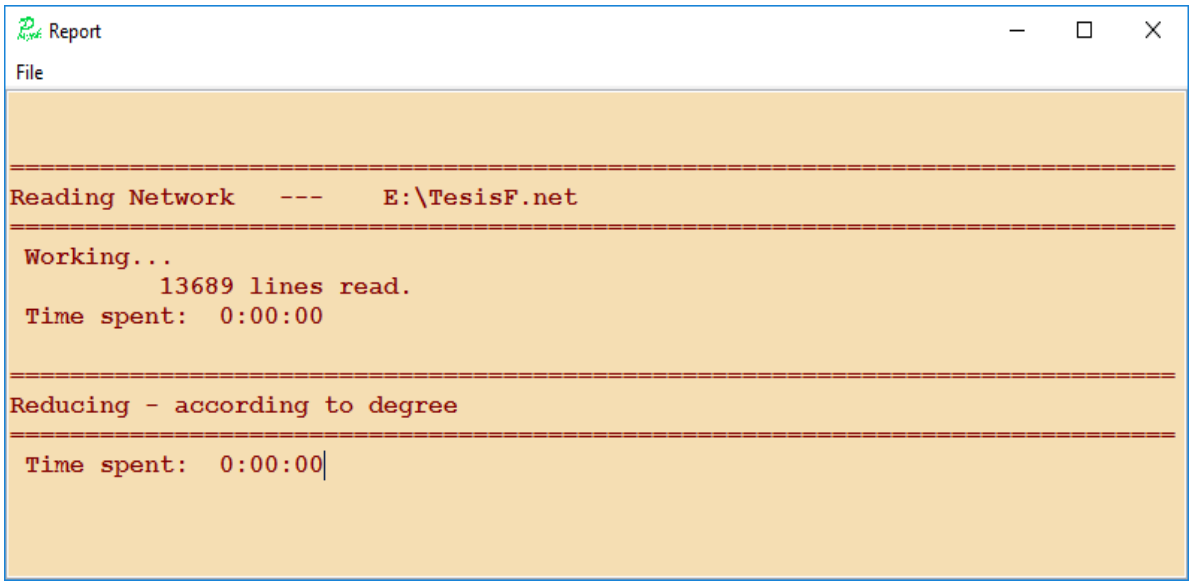


Figura 4. 14. Reporte del archivo Tesis.net simplificado

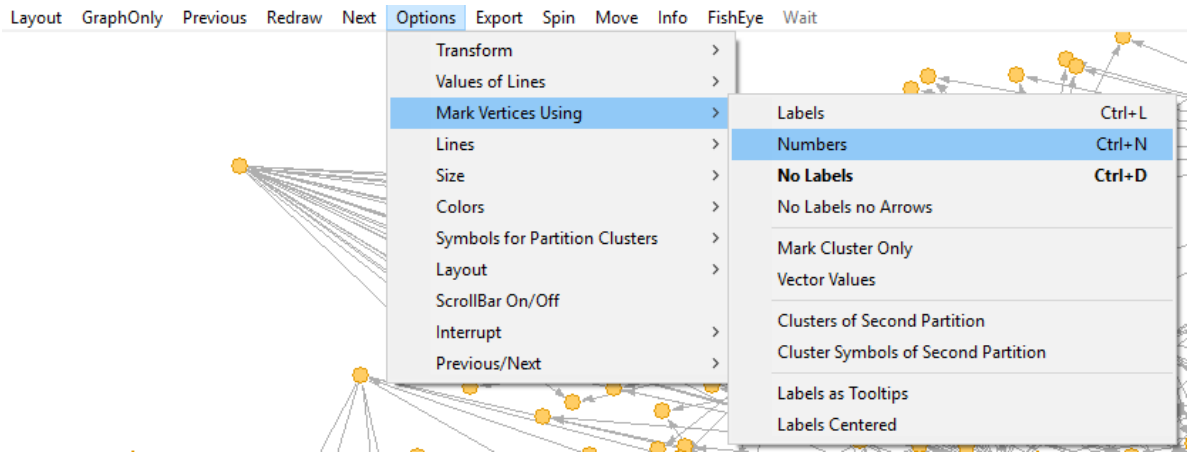


Figura 4. 15. Opciones *Options/ Mark Vertices Using /Numbers*

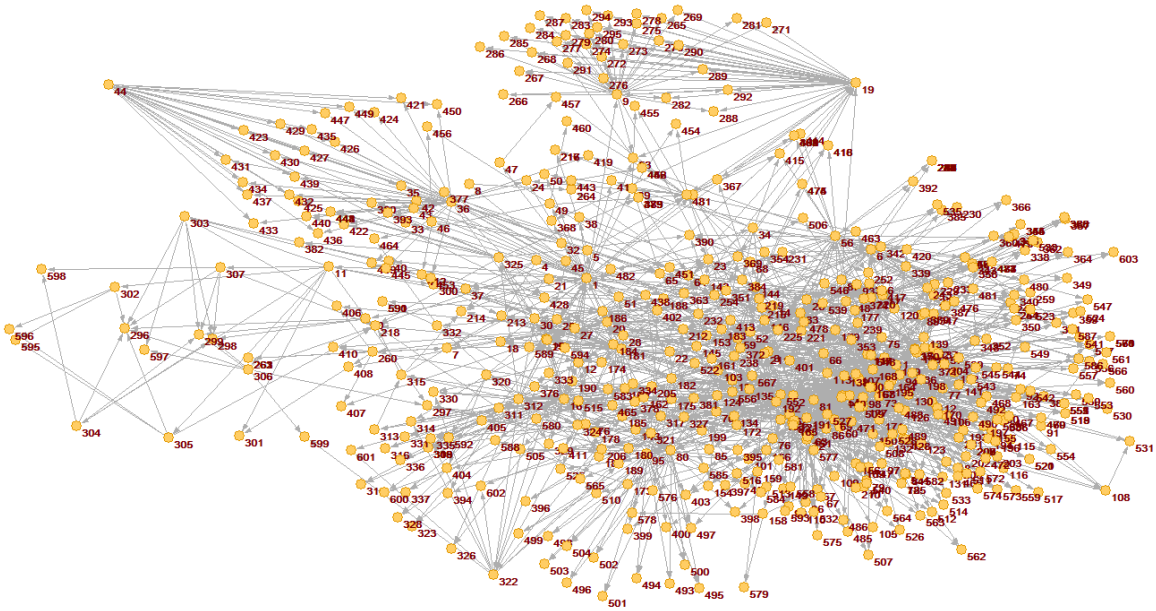


Figura 4. 16. Grafo del archivo *Tesis.net* simplificado

- Guardamos la red simplificada, para su posterior uso en la construccion de la matriz de adyacencia . Para tal fin hacemos clic botón *Guardar* (recuadro rojo) en la seccion *Networks* (figura 4.17). El archivo tendra el nombre *TesisFinal.net*. Automaticamente nos abrira una ventana "*Report*" indicando que la exportación fue exitosa. Ver figura 4.18

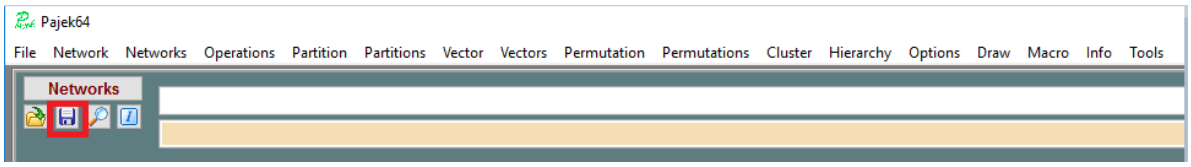


Figura 4. 17. Ventana para guardar el grafo simplificado

```
Report
File
==============================================================================
 Working...
          13689 lines read.
 Time spent:  0:00:00

==============================================================================
Reducing - according to degree
==============================================================================
 Time spent:  0:00:00

==============================================================================
Saving network to file   ---    C:\Users\Administrador\Documents\TesisFinal.net
==============================================================================
 Time spent:  0:00:00
```

Figura 4. 18. Reporte del archivo TesisFinal.net

- Al abrir el archivo *TesisFinal.net* tendremos el correspondiente formato. Ver figura 4.19 y 4.20.

```
TesisFinal.net: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
*Vertices 603
   1 "www.unapiquitos.edu.pe"          0.5987    0.5484    0.5000
   2 "www.peru.gob.pe"                 0.3499    0.6122    0.5000
   3 "issuu.com"                       0.4471    0.6467    0.5000
   4 "revistas.unapiquitos.edu.pe"     0.6166    0.5213    0.5000
   5 "www.escritores.org"              0.6337    0.5567    0.5000
   6 "www.un.org"                      0.5076    0.7845    0.5000
   7 "agenda.pucp.edu.pe"              0.5872    0.3932    0.5000
   8 "jooble.pe"                       0.7799    0.5297    0.5000
   9 "empleo.universia.edu.pe"         0.8471    0.7089    0.5000
  10 "dspace.unapiquitos.edu.pe"       0.6533    0.3498    0.5000
```

Figura 4. 19. Lista de los vértices del archivo *TesisFinal,net*

```
TesisFinal.net: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
*Arcs
   1    2 1
   1    3 1
   1    4 1
   1    5 1
   1    6 1
   1    7 1
   1    8 1
   1    9 1
   1   10 1
   1   11 1
```

Figura 4. 20. Lista de los Arcs del archivo *TesisFinal.net*

- Si deseamos medir la centralidad (importancia o poder un nodo(vertice)), debemos ir a la opcion ***Network/Create Vector/Centrality/Weighted Degree/All***. Ver figura 4.21

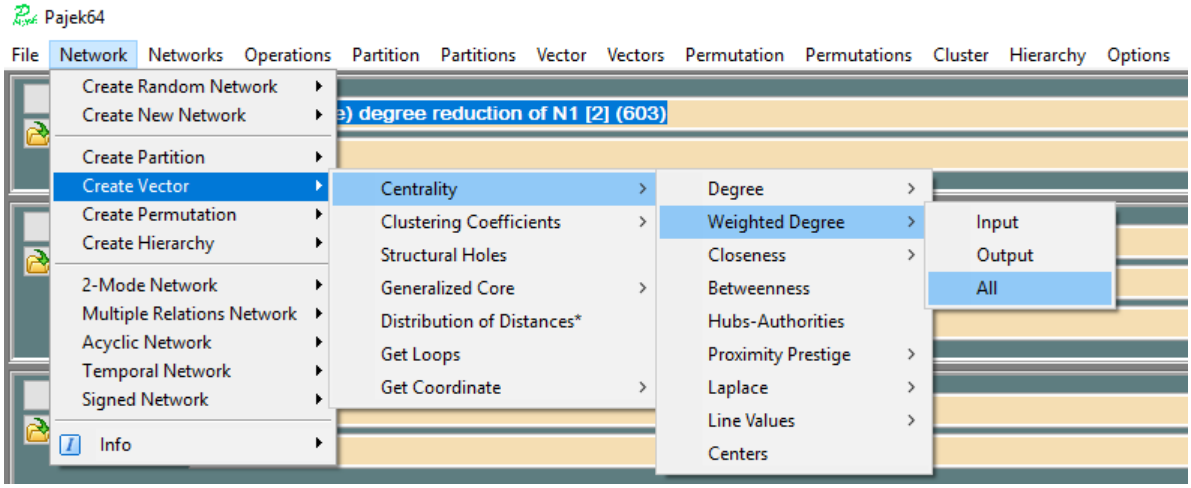


Figura 4. 21. Opciones ***Network/Create Vector/Centrality/Weighted Degree/All***

- El nuevo grafo con centralidad se muestra en la figura 4.22, como observamos los nodos con mayor tamaño son los que tienen mas importancia o relevancia.

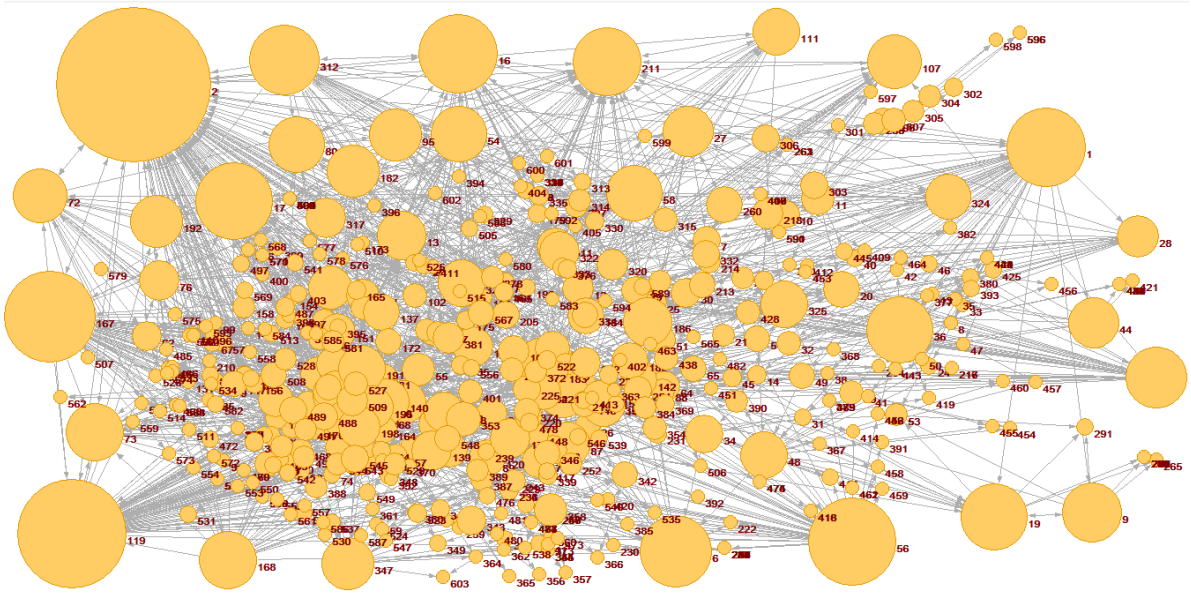


Figura 4. 22. Grafo del archivo *TesisFinal.net* con centralidad

4.4.CONSTRUCCIÓN DE LA MATRIZ DE ADYACENCIA

Con el archivo *TesisFinal.net*, importaremos a Matlab los elementos para la construccion de la matriz de adyacencia, para tal fin construimos y ejecutamos los *archivos.m*.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Este programa extrae una lista de Arcs de un archivo de texto pajek (.net)
%============================================================================
% Entrada: Ruta de archivo.net, número de nodos en el grafico
% Salida: Lista de Arcs
% Autor: MIT Strategic Engineering Matlab Tools for Network Analysis(2006-2011)
% Disponible:http://strategic.mit.edu/downloads.php?page=matlab_networks
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function el=pajek2edgeL(filename,n)

[e1,e2,e3] = textread(filename,'%6d%6d%6d','headerlines',n+2);
el=[e1,e2,e3];



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Convierte la lista de Arcs en la matriz de adyacencia
% ====================================================
% Entrada: Lista de Arcs (generado por la función pajek2edgeL)
% Salida : matriz de adyacencia nxn
% Autor: MIT Strategic Engineering Matlab Tools for Network Analysis(2006-2011)
% Disponible:http://strategic.mit.edu/downloads.php?page=matlab_networks
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function adj=edgeL2adj(el)

nodes=sort(unique([el(:,1) el(:,2)])); % Obtener todos los nodos ordenados
adj=zeros(numel(nodes));    % Inicializar matriz de adyacencia

% Recorriendo todos los Arcs
for  i=1:size(el,1);  adj(find(nodes==el(i,1)),find(nodes==el(i,2)))=el(i,3);
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Este programa extrae una matriz de adyacencia de un archivo pajek (.net).
% Además extrae en un vector columna, los valores numéricos de cada nodo(vértice)
% ========================================================================
% Entrada: Ruta del archivo .net , número de nodos en el grafico
% Salida : matriz de adyacencia nxn
% Funciones invocadas: pajek2edgeL.m, edgeL2adj.m
% Autor: Julio Martin Rojas Tenazoa
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [adj,Nodos] = pajek2adj(filename,n)

el=pajek2edgeL(filename,n);
adj=edgeL2adj(el);
[number,url, valor1,valor2,valor3] = textread(filename,'%d %s %f %f
%f',n,'headerlines',1);
Nodos=number;
```

Enseguidamente generaremos la matriz de adycencia y el vector columna con los valores numericos de cada nodo, para tal proposito en el prompt o línea de comandos del sistema ($fx \gg$) , digitamos la función *pajek2adj.m*, referenciando la ruta del archivo .net y la cantidad de nodos, como parámetros. Como muestra el código:

```
fx >> [M,Node] = pajek2adj('C:\TesisFinal.net',603)
```

Al ejecutar, almacenará en la variable M y *Node* , la matriz adyacente de tamaño 603x603 y el vector columna con los valores de cada nodo, que es formada a partir del archivo *TesisFinal.net*

4.5.ASIGNACION DEL PAGERANK DE CADA NODO(VERTICE)

Enseguida, mediante la matriz M , que es la matriz adyacente de tamaño 603x603, se formara el vector de importancias o PageRank de cada uno de los nodos(vertices) implicados, para tal fin se implemento la funcion *Pagerank.m* , que recibe como parametro la matriz M y *Node*, y por medio del método de las potencias, devuelve el vector de importancias.

```
%**********************************************************************
% Calculo del PageRank mediante el método de las potencias
% =========================================================
% Función invocada: MatrizEstoFilas.m
% Ultima fecha de depuración de errores: 21 de marzo del 2017
% Autor: Julio Martin Rojas Tenazoa
%**********************************************************************
function [ValorP,VectorP,Delta,itera]= PageRank(A,Nodo)
[m n] = size(A);
if m==n
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    M=double(full(A)');
    e=ones(n,1);
    v=(1/n)* e;
    A=0.85*MatrizEstoFilas(M)+(1-0.85)*e*v';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    x0=diag(A'*A);
    x0=x0/norm(x0);

    x1 = (A'*A)*x0;
    x1 = x1/norm(x1);

    x2 = (A'*A)*x1;
    x2 = x2/norm(x2);

    lambda = x1'*x0;
    lambda2 = x2'*x1;

    ValorP=lambda2;
    VectorP=x2;

    Delta=abs((lambda2-lambda)/(lambda2));
    itera=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
while 1.0e-16 < Delta
    x = (A'*A)*x2;
    x = x/norm(x);
    lambdanew =x'*x2;
    Delta=abs((lambdanew-ValorP)/(lambdanew));
    ValorP = lambdanew;
    VectorP=x;
    x2= x;
    itera=itera+1;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Page=[Nodo,VectorP];
[B,k] = sort(Page(:,2),'descend');
B = [Page(k) B];
disp('    Nodos      PageRank');disp('');disp(B)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Este programa convierte una matriz de adyacencia, en una matriz
% estocástica o de Markov (estocásticamente por filas)
% ==========================================================================
% Entrada: Matriz de adyacencia nxn
% Salida : Matriz estocástica o de Markov
% Autor: Julio Martin Rojas Tenazoa
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function S = MatrizEstoFilas (M)
link_usc = sum (M,2);
n = length (M);
S = M;
for i = 1:n
if link_usc(i) == 0
S(i,:) = 1 / n;
else
S(i,:) = S(i,:) / link_usc(i);
end
end
```

Seguidamente generamos el vector de importancias o Pagerank, para tal proposito en el prompt o línea de comandos del sistema ($fx \gg$), digitamos la función *PageRank.m*, referenciando la matriz adyacente *M* y el vector columna *Node*. Como muestra el código:

```
fx >>[ValorP,VectorP,Delta,itera]= PageRank(M,Node)
```

Al ejecutar, almacenara en *ValorP* al valor propio de máximo modulo en las n iteraciones efectuadas, en *VectorP* almacenara el vector de importancia o PageRank en las n iteraciones efectuadas, en *Delta* almacenara el error relativo y finalmente en *itera* almacenara la cantidad de iteraciones. Cabe mencionar que devolvera el vector de importancias de cada nodo ordenada de mayor a menor. Como muestra la lista.

Nodos	PageRank
2.0000	0.9883
119.0000	0.1278
167.0000	0.0432
181.0000	0.0286
54.0000	0.0240
312.0000	0.0199
16.0000	0.0199
27.0000	0.0199
130.0000	0.0164
168.0000	0.0158
73.0000	0.0157
72.0000	0.0139
139.0000	0.0106
56.0000	0.0096
77.0000	0.0094
58.0000	0.0088
191.0000	0.0087
182.0000	0.0085
17.0000	0.0084
347.0000	0.0077
192.0000	0.0071
82.0000	0.0070
80.0000	0.0069
126.0000	0.0067
137.0000	0.0067
124.0000	0.0063
152.0000	0.0058
127.0000	0.0058
322.0000	0.0054
121.0000	0.0050
321.0000	0.0050
164.0000	0.0048
6.0000	0.0044
28.0000	0.0043
144.0000	0.0037
133.0000	0.0037
177.0000	0.0036
198.0000	0.0034
86.0000	0.0034
138.0000	0.0034
112.0000	0.0033
317.0000	0.0032
186.0000	0.0030
95.0000	0.0028
107.0000	0.0028
140.0000	0.0028
1.0000	0.0027
111.0000	0.0027
108.0000	0.0027
158.0000	0.0026

El vector de importancias mostrada en la pagina anterior (solo tomamos los 50 primeras de 603), convergio en 28 iteraciones, dando como valor propio el valor de 1, tal como lo predice el teorema de Perron-Frobenius.

A continuación, en la tabla 4.1, se listara las 50 páginas con mejor PageRank; la tabla contiene en el orden especifico, los nodos numerados, las páginas URL y los valores del PageRank de cada nodo .

Tabla 4. 1. Lista de las 50 páginas con mejor PageRank

Nodos	URL	PageRank
2	www.peru.gob.pe	0.9883
119	www.minsa.gob.pe	0.1278
167	www.defensoria.gob.pe	0.0432
181	www.uncp.edu.pe	0.0286
54	www.ongei.gob.pe	0.0240
312	iiap.org.pe	0.0199
16	www.iiap.org.pe	0.0199
27	rpu.edu.pe	0.0199
130	www.agrobanco.com.pe	0.0164
168	www.fpcmac.org.pe	0.0158
73	www.mtc.gob.pe	0.0157
72	www.minem.gob.pe	0.0139
139	www.bcrp.gob.pe	0.0106
56	www.oecd.org	0.0096
77	www.elperuano.com.pe	0.0094
58	sgrd.pcm.gob.pe	0.0088
191	www.regionhuanuco.gob.pe	0.0087
182	www.unheval.edu.pe	0.0085

Nodos	URL	PageRank
17	www.minedu.gob.pe	0.0084
347	www.pronied.gob.pe	0.0077
192	www.regionamazonas.gob.pe	0.0071
82	www.pepp.gob.pe	0.0070
80	www.inia.gob.pe	0.0069
126	www.ccffaa.mil.pe	0.0067
137	ddclalibertad.gob.pe	0.0067
124	www.osinfor.gob.pe	0.0063
152	www.cultura.gob.pe	0.0058
127	www.ejercito.mil.pe	0.0058
322	siamazonia.iiap.org.pe	0.0054
121	www.inabif.gob.pe	0.0050
321	www.promamazonia.org.pe	0.0050
164	www.amag.edu.pe	0.0048
6	www.un.org	0.0044
28	www.cienciactiva.gob.pe	0.0043
144	www.uni.edu.pe	0.0037
133	www.inaigem.gob.pe	0.0037
177	www.unac.edu.pe	0.0036
198	www.regionayacucho.gob.pe	0.0034
86	www.sutran.gob.pe	0.0034
138	www.juntos.gob.pe	0.0034
112	www.hospitalchulucanas.gob.pe	0.0033
317	www.sisociodiversidad.org.pe	0.0032
186	www.unmsm.edu.pe	0.0030
95	inia.gob.pe	0.0028
107	portal.osce.gob.pe	0.0028

Nodos	URL	PageRank
140	www.mpfn.gob.pe	0.0028
1	www.unapiquitos.edu.pe	0.0027
111	www.essalud.gob.pe	0.0027
108	diariooficial.elperuano.pe	0.0027
158	www.munisjm.gob.pe	0.0026

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

Google sin duda es la pagina mas visitada por millones de personas, por la cual requiere de algoritmos como el PageRank para brindar un optimo servicios a los millones de internautas y proporcionar resultados de busquedas con informacion requerida.

PageRank ha sido el centro de google, pero no es el unico algoritmo que utiliza para clasificar las paginas de acuerdo a su importancia. Sin embargo es el mas conocido.

El exito del PageRank, se debe a las matematicas en especial a la cadena de Markov y al teorema de Perron-Frobenius, puesto que la matriz utilizada por Google tambien llamada matriz Google, tiene propiedades de ser una matriz estocastica, una matriz no negativa e irreductible,la cual es un vector propio asociado al valor propio 1; gracias a que cumple con todas las propiedades ya mencionadas; podemos estar seguros que el vector de importancias o PageRank, es un optimo algoritmo para el ordenamiento de las paginas web, ya que el teorema de Perron-Frobenius nos aseguran que este vector existe y es unico.

Al terminar el presente trabajo de tesis, se concluye que se ha cumplido con el objetivo general que comprendía en aproximar matemática y computacionalmente al funcionamiento del motor de búsqueda Google.

5.2. RECOMENDACIONES

Que los docentes de la FISI-UNAP implicados en cursos como algebra Lineal, calculo diferencia e integral, ecuaciones diferenciales, métodos numéricos u otros afines, complementen en el desarrollo de sus clases, la elaboración de algoritmo con modelos matemáticos, como además la construcción de la misma en Matlab u otro software de entorno matemático.

Enfatizar en los alumnos y egresados de la FISI-UNAP, la importancia que tiene hoy en día la matemática computacional en el desarrollo de tecnologías recientes y fomentar la realización de estudios empleando a la simulación como herramienta para analizar el comportamiento de diferentes modelos matemáticos.

GLOSARIO DE TÉRMINOS

Término	Significado
Algoritmo	Es un conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite llevar a cabo una actividad mediante pasos sucesivos que no generen dudas a quien deba hacer dicha actividad.
Algebra lineal	Rama de las matemáticas que se encarga de estudiar los conceptos tales como vectores, matrices, sistemas de ecuaciones lineales, espacios vectoriales, y transformaciones lineales.
Aristas	En teoría de grafos, una arista corresponde a una relación entre dos vértices de un grafo.
Base de datos	Una base de datos (cuya abreviatura es BD) es una entidad en la cual se pueden almacenar datos de manera estructurada, con la menor redundancia posible. Diferentes programas y diferentes usuarios deben poder utilizar estos datos.
Buscador	Un buscador es un sistema informático que nos permite encontrar páginas web o resultados en base a la frase o palabra que hayamos ingresado y estemos buscando.
Cadenas de Markov	Una cadena de Markov es una serie de eventos, en la cual la probabilidad de que ocurra un evento depende del evento inmediato anterior. En efecto, las cadenas de este tipo tienen memoria, "recuerdan" el último evento y esto condiciona las posibilidades de los eventos futuros.
Capital de riesgo	Es una forma de financiar empresas que están naciendo y que no tienen un historial que permita confiar en sus resultados o tener la seguridad de que se recibirán retornos por el dinero que se le preste.
Combinación lineal	En matemática, particularmente en álgebra lineal, una combinación lineal es una expresión matemática que consiste en la suma entre pares de elementos, de determinados conjuntos, multiplicados entre sí.

Término	Significado
Diagonal principal	En álgebra lineal, la diagonal principal de una matriz cuadrada contiene los elementos situados desde $a_{1,1}$ hasta $a_{n,n}$. Es decir, los elementos que van desde la esquina superior izquierda hasta la esquina inferior derecha: $a_{1,1}$, $a_{2,2}$, $a_{3,3}$ … $a_{n,n}$
Distribución de probabilidad	En teoría de la probabilidad y estadística, la distribución de probabilidad de una variable aleatoria es una función que asigna a cada suceso definido sobre la variable aleatoria la probabilidad de que dicho suceso ocurra.
Escalar	En términos matemáticos, se llama escalar a los elementos de un cuerpo , generalmente números, y en particular se usa cuando los vectores en el álgebra lineal y en cualquier rama que use módulos o espacios vectoriales.
Espacio vectorial	En álgebra abstracta, un espacio vectorial es una estructura algebraica creada a partir de un conjunto no vacío, una operación interna (llamada suma, definida para los elementos del conjunto) y una operación externa (llamada producto por un escalar, definida entre dicho conjunto y otro conjunto, con estructura de cuerpo)
Google AdWords	Es un servicio y un programa de la empresa Google que se utiliza para ofrecer publicidad patrocinada a potenciales anunciantes.
Grafo	Un grafo es un conjunto, no vacío, de objetos llamados vértices (o nodos) y una selección de pares de vértices, llamados aristas (edges en inglés) que pueden ser orientados o no. Típicamente, un grafo se representa mediante una serie de puntos (los vértices) conectados por líneas (las aristas).
Grafo dirigido	Un grafo dirigido o digrafo es un tipo de grafo en el cual las aristas tienen un sentido definido.

Término	Significado
IDE	Un IDE es una herramienta que nos ayuda a desarrollar de una manera amigable nuestras aplicaciones, brindándonos ayudas visuales en la sintaxis, plantillas, wizards, plugins y sencillas opciones para probar y hacer un debug. Programas como Netbeans, Eclipse, Visual Studio, Matlab, son algunos de los IDE's más populares.
Índice	El índice de una base de datos es una estructura de datos que mejora la velocidad de las operaciones, por medio de identificador único de cada fila de una tabla, permitiendo un rápido acceso a los registros de una tabla en una base de datos.
Internauta	Persona que utiliza los servicios de internet u otra red informática.
Javascript	JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.
Matriz cuadrada	Una matriz de n por m elementos, es una matriz cuadrada si el número de filas es igual al número columnas, es decir, $n = m$ y se dice, entonces que la matriz es de orden n.
Matriz de adyacencia	Matriz cuadrada de orden nxn asociada a un grafo de orden n, donde sus filas y columnas se identifican con los vértices del grafo y en las celdas se indican la cantidad de aristas (o arcos salientes si es un dígrafo) a los nodos asignado a la fila y columnas en cuestión.
Matriz de permutación	La matriz permutación es la matriz cuadrada con todos sus nxn elementos iguales a 0, excepto uno cualquiera por cada fila y columna, el cual debe ser igual a 1.

Término	Significado
Matriz estocástica	En matemáticas, una matriz estocástica (también denominada matriz de probabilidad, matriz de transición, matriz de sustitución o matriz de Markov) es una matriz utilizada para describir las transiciones en una cadena de Markov.
Matriz estocástica por columnas o por la izquierda	Es una matriz cuadrada, donde cada una de cuyas columnas está formada por números reales no negativos, sumando cada columna 1.
Matriz Google	Es una matriz estocástica que es utilizado por el algoritmo de Google PageRank . La matriz es un gráfico con los bordes que representan los enlaces entre páginas. La clasificación de cada página se puede generar de forma iterativa a partir de la matriz de Google utilizando el método de la potencia . Sin embargo, con el fin de que el método de la potencia a converger, la matriz debe ser estocástico , irreducible y aperiódica.
Matriz idempotente	Una matriz idempotente es una matriz la cual es igual a su cuadrado, es decir A es idempotente si $A^2 = A$
Matriz Identidad	En álgebra lineal, la matriz identidad es una matriz que cumple la propiedad de ser el elemento neutro del producto de matrices. Esto quiere decir que el producto de cualquier matriz por la matriz identidad (donde dicho producto esté definido) no tiene ningún efecto. Además es una matriz diagonal en la que los elementos de la diagonal principal son iguales a 1.
Matriz ortogonal	Es una matriz cuadrada cuya matriz inversa coincide con su matriz traspuesta. decir A es ortogonal si $A^{-1} = A^t \Rightarrow A^{-1}.A^t = I$.
Matriz simétrica	Una matriz es simétrica si es una matriz cuadrada, la cual tiene la característica de ser igual a su traspuesta. Es decir A es simétrica si $A = A^t$.

Término	Significado
Matriz singular	Matriz cuadrada cuyo determinante es igual a cero. Una matriz singular no tiene matriz inversa.
Matriz transpuesta	Dada una matriz A, se llama matriz transpuesta de A a la matriz que se obtiene cambiando ordenadamente las filas por las columnas.
Matriz triangular	En álgebra lineal, una matriz triangular es un tipo especial de matriz cuadrada cuyos elementos por encima o por debajo de su diagonal principal son cero.
Método iterativo	Un método iterativo trata de resolver un problema matemático (como una ecuación o un sistema de ecuaciones) mediante aproximaciones sucesivas a la solución, empezando desde una estimación inicial.
Métodos de las potencias	En análisis numérico, el método de las potencias es un método iterativo que calcula sucesivas aproximaciones a los vectores propios y valores propios de una matriz. El método se usa principalmente para calcular el vector propio de mayor valor propio en matrices grandes.
Open Source	Open Source (Código abierto) es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones éticas y morales las cuales destacan en el llamado software libre.
Parseo	Proceso de analizar una secuencia de símbolos a fin de determinar su estructura gramatical con respecto a una gramática formal dada. Formalmente es llamado análisis de sintaxis. Un parseador (parser) es un programa de computación que lleva a cabo esta tarea. El parseo transforma una entrada de texto en una estructura de datos (usualmente un árbol) que es apropiada para ser procesada. Generalmente los parseadores primero identifican los símbolos de la entrada y luego construyen el árbol de parseo para esos símbolos.

Término	Significado
Path	Es una variable de entorno de los sistemas operativos POSIX y los sistemas de Microsoft, en ella se especifican las rutas en las cuales el intérprete de comandos debe buscar los programas a ejecutar.
Polinomio característico	En álgebra lineal, se asocia un polinomio a cada matriz cuadrada llamado polinomio característico. Dicho polinomio contiene una gran cantidad de información sobre la matriz, los más significativos son los valores propios, su determinante y su traza.
Probabilidad	La probabilidad es la mayor o menor posibilidad de que ocurra un determinado suceso. En otras palabras, su noción viene de la necesidad de medir o determinar cuantitativamente la certeza o duda de que un suceso dado ocurra o no.
Probabilidad condicional	Es la probabilidad de que ocurra un evento A, sabiendo que también sucede otro evento B. La probabilidad condicional se escribe $P(A\|B)$, y se lee "la probabilidad de A dado B".
Probabilidad de transición	En la teoría de los procesos estocásticos y, en particular, en la teoría de las cadenas de Markov, se denomina probabilidad de transición, a la probabilidad de que estando el sistema en el estado E_i en el momento n pase al estado E_i en el momento $n + 1$.
Proceso estocástico	Un proceso estocástico es un fenómeno aleatorio que surge en un proceso que se desarrolla en el tiempo de una manera controlada por medio de leyes probabilísticas.
Redes sociales	Una red social es una estructura social compuesta por un conjunto de actores (tales como individuos u organizaciones) que están relacionados de acuerdo a algún criterio (relación profesional, amistad, parentesco, etc.). Normalmente se representan simbolizando los actores como nodos y las relaciones como líneas que los unen.

Término	Significado
Simulación	En las ciencias, la simulación es el artificio contextual que referencia la investigación de una hipótesis o un conjunto de hipótesis de trabajo utilizando modelos.
Sin pérdida de generalidad	Es una expresión utilizada en las demostraciones matemáticas y que introduce una suposición particular, de tal manera que el caso general pueda mostrarse que es equivalente a ese caso particular. Cuando esto sucede, dicha suposición o elección particular es irrelevante para la demostración, y presenta la ventaja de que permite reducir la extensión de la demostración reduciendo el número de casos que hay que analizar.
Traza de una matriz	Se define la traza de la matriz A y se denota por $traza(A)$ al valor obtenido al sumar todos los elementos de la diagonal principal, es decir $$traza(A) = \sum_{i=1}^{n} a_{ii}$$
Urls	URL es una sigla del idioma inglés correspondiente a *Uniform Resource Locator (Localizador Uniforme de Recursos)*. Se trata de la secuencia de caracteres que sigue un estándar y que permite denominar recursos dentro del entorno de Internet para que puedan ser localizados.
Vector	Segmento de recta, contado a partir de un punto del espacio, cuya longitud representa a escala una magnitud, en una dirección determinada y en uno de sus sentidos.
Vector de probabilidad	Es un vector en la que las entradas son no negativa y agregadas hasta 1. Las entradas en un vector probabilidad pueden representar las probabilidades de encontrar un sistema de cada uno de los estados.
Vector Nulo	Un vector nulo o vector cero se refiere a un vector que posee módulo (o extensión) nulo.

Término	Significado
Vértices o nodos	En teoría de grafos, un vértice o nodo es la unidad fundamental de la que están formados los grafos. Un grafo no dirigido está formado por un conjunto de vértices y un conjunto de aristas (pares no ordenados de vértices), mientras que un grafo dirigido está compuesto por un conjunto de vértices y un conjunto de arcos (pares ordenados de vértices).

BIBLIOGRAFÍA

[1] PAPROTUL 1001., "Proceso estocástico I", Universidad Nacional Autónoma de México, 2014. Disponible: http://lya.fciencias.unam.mx/lars/0630/

[2] Torres G. M., "Cadenas de Markov aplicadas al ordenamiento de páginas web", Tesis presentada como parte de los requisitos para optar el grado de Licenciada en Matemáticas Aplicadas, Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias Físico Matemáticas, 2016, pág.6. Disponible: http://www.fcfm.buap.mx/assets/docs/docencia/tesis/ma/MariaGuadalupeTorresGonzalez.pdf

[3] Marín D. J., "Cadenas de Markov", Universidad Carlos III de Madrid., 2014, pág. 28. Disponible: http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/PEst/tema4pe.pdf

[4] "Espacios Normados (Normas en $\mathbb{R}^n$)", Universidad Nacional Autónoma de México, 2016, pág. 1. Disponible: http://sistemas.fciencias.unam.mx/~erhc/espacios%20normados.pdf

[5] "Elementos del Algebra Lineal. Normas", Universidad de Sevilla, 2016, pág. 5. Disponible: http://departamento.us.es/edan/php/asig/LICMAT/LMCN2/ApuntesCN20809.pdf.

[6] Coutiño A. D. y Maximenko E., "Norma de Frobenius", Instituto Politécnico Nacional Escuela Superior de Física y Matemáticas, 2016, págs. 1 y 8. Disponible: http://esfm.egormaximenko.com/linalg/Frobenius_norm_es.pdf.

[7] Díez M. M., "Matrices, aplicaciones lineales y métodos directos de resolución de sistemas de ecuaciones lineales", Grupo de Dinámica de Flujos Ambientales Universidad de Granada, 2015, pág. 11. Disponible: http://wdb.ugr.es/~mdiezm/wp-content/uploads/Tema-6-v0.2.pdf

[8] "Dependencia e Independencia lineal". Disponible: https://es.wikipedia.org/wiki/Dependencia_e_independencia_lineal.

[9] "Diagonalización de Matrices", Universidad de Cordoba, 2015, pág. 2. Disponible: http://www.uco.es/geometria/documentos/Tema4Biologia_Diagonalizacion.pdf

[10] "Autovalores y Autovectores", Universidad de Sevilla, 2014, págs 1-3. Disponible: http://personal.us.es/ealgaba/algebra%20=04-05/TEMA11.pdf.

[11] Gómez G. F., "El Teorema de Cayley-Hamilton", Universidad de Murcia, 2015, pág. 1. Disponible: http://www.uv.es/asepuma/XII/comunica/gomez.pdf

[12] "Diagonalización de Matrices", Universidad de Sevilla, 2016, pág. 1-3. Disponible: http://personal.us.es/jsmonter/jes1/pdf/DiagMarilin.pdf

[13] "La forma canónica de matrices y endomorfismos", Universidad Autónoma de Madrid, Departamento de Matematicas,2015, pág 51. Disponible: http://verso.mat.uam.es/~andrei.jaikin/algebralineal2007-2008/tema5.pdf

[14] "Multiplicidad Algebraica y Geométrica", Universidad de Sevilla, Repositorio de Objetos de Aprendizaje de la Universidad de Sevilla (RODAS), 2016. Disponible: http://personal.us.es/jsmonter/jes1/pdf/DiagMarilin.pdf

[15] "Endomorfismos y Diagonalización", Universidad de Cantabria, 2015, págs. 8-10. Disponible: http://ocw.unican.es/ensenanzas-tecnicas/algebra-lineal-y-geometria/materiales-de-clase1/Tema6_Endomorfismos%20y%20Diagonalizacion.pdf.

[16] "Autovalores y autovectores de una matriz cuadrada", Universidad de La Laguna, 2014, págs. 1-3. Disponible: http://www.ea.ull.es/asig/mat2/TAAUTOV.PDF

[17] "Propiedades de los autovalores de una matriz", Universidad de Sevilla 2012, págs. 1-2. Disponible:
http://torricelli.uvigo.es/Dropbox/apuntes_de_clase/Algebra_Lineal/handouts/cap_5/propiedades.pdf

[18] "Valores propios y vectores propios", Universidad de Sevilla 2012, págs. 114. Disponible: http://www.etsist.upm.es/uploaded/docs_personales/hernandez_heredero_rafael_jose/old/Algebra/ALApC6.pdf

[19] Adrián I. A., "El Teorema de Perron-Frobenius y su aplicación en el algoritmo de búsqueda de Google", Tesis presentada como parte de los requisitos para optar el grado de Matemáticas, Universidad de la Rioja, Facultad de Ciencias, Agroalimentarios e Informática, 2015, págs.5, 21-31. Disponible:
http://biblioteca.unirioja.es/tfe_e/TFE001026.pdf

[20] Fernández G. P., "El secreto de Google y el Álgebra lineal", Universidad Autónoma de Madrid, Departamento de Matemáticas, 2011, págs.9-10.
Disponible: https://www.uam.es/personal_pdi/ciencias/gallardo/google_sema.pdf

[21] "Autovalor de Perrón. Aplicaciones", Federación de Enseñanza de Comisiones Obreras de Andalucía, 2012, págs.9. Disponible: https://www.feandalucia.ccoo.es/docu/p5sd4864.pdf

[22] "Método de las potencias", FdeT, 2015. Disponible: http://fdet.es/

[23] "Araña web". Disponible: https://es.wikipedia.org/wiki/Ara%C3%B1a_web

[24] Coso S. A., "Desarrollo de Infraestructuras para el Modelado de Usuarios", Tesis presentada como parte de los requisitos para optar el título de Ingeniería Informática, Universidad Carlos III de Madrid, Escuela Politécnica Superior, 2009, págs.8-10.

Disponible:http://orff.uc3m.es/bitstream/handle/10016/8544/PFC_Ana_Coso_Santos.pdf;jsessionid=F7E50206B6397B629AC64CD42BEC060C?sequence=1

[25] Maturana M. P., Cornejo. F., "El Estudio de Sistemas de Ciudades. una Aproximación desde Métodos de Análisis de Redes Sociales", Universidad de la Frontera, 2010, pág. 5. Disponible: http://www.ubiobio.cl/miweb/webfile/media/222/Espacio/2010/EL%20ESTUDIO%20DE%20SISTEMAS%20DE%20CIUDADES.pdf

[26] "Pajek Εφαρμογή ανάλυσης μεγάλων δικτύων Έκδοση 4.03", pág. 2. Disponible: https://pithos.okeanos.grnet.gr/public/NP6pYs83IZT4phhxmPlXJ4

[27] HO-450 Verão 2013: Pajek., Material baseado nos capítulos 2, 3 e 6 do livro ESNA, 2011. Nooy, Mrvar & Batagelj, Ed. Cambridge, como material auxiliar na disciplina do curso de verão HO-450, ministrada em 2013 no Instituto de Economia da Universidade Estadual de Campinas., págs.9-10.
Disponible: http://mrvar.fdv.uni-lj.si/pajek/portuguese/portuguese.pdf

[28] Ruiz L. A., Jung N. I., "Visualización con "Pajek"", Laboratorio de Redes, IIMAS, Universidad Nacional Autónoma de México, 2013, págs. 9-17. Disponible: http://www.academia.edu/9578799/Visualizaci%C3%B3n_con_Pajek_

[29] López M. M., Acosta R. J.A., "Manuel de Introducción al Matlab", Universidad de Sevilla 2014, págs. 5 y 6. Disponible: http://www.esi2.us.es/~jaar/Datos/RegAuto/Practica1.pdf

[30] "MATLAB". Disponible: https://es.wikipedia.org/wiki/MATLAB

[31] MaTLaB, "Entorno de Trabajo". Disponible: http://matlabxsiemprexd.blogspot.pe/2007/11/entorno-de-trabajo.html

[32] "El entorno MATLAB". Universidad de Sevilla, pág. 3. Disponible: http://www.esi2.us.es/~jaar/Datos/FIA/Enunciados.pdf

[33] Saldarriaga O., Giraldo H., "Algebra Lineal con el uso de Matlab", Instituto de Matemáticas, Universidad de Antioquia, 2010, págs. 192-195. Disponible: http://aprendeenlinea.udea.edu.co/lms/moodle/pluginfile.php/199019/mod_forum/attachment/174141/Algebra_lineal.pdf

[34] "Introducción a la Programación en Matlab", págs. 1-4. Disponible: http://www.galeon.com/algebralineal/algii_inform/int_prog_matlab.pdf

[35] Lamarca L. M J., "Buscadores". Disponible: http://www.hipertexto.info/documentos/buscar.htm

[36] Ezquerro M. L. M., "Cómo encontrar una aguja en un pajar", Lección inaugural del curso académico 2013-2014, Universidad Pública de Navarra, 2013, págs 8,10-12. Disponible: https://www.unavarra.es/digitalAssets/173/173530_100000Leccion-inaugural-Castellano_FINAL.pdf

[37] "Significado de Google". Disponible:https://www.significados.com/google/

[38] Carreras L. R., "Cómo clasifica Google los resultados de las búsquedas: factores de posicionamiento orgánico", Tesis presentada como parte de los requisitos para optar el grado de Doctor, Universidad Complutense de Madrid, Facultad de Ciencias de la Información, Departamento de Periodismo III (Teoría General de la Información), 2012, págs. 16-18. Disponible: http://eprints.ucm.es/17450/

[39] Balderas A. L., Salvador M. D., Gonzalo M. M., Chaparro S. R., "Aplicación del algoritmo de PageRank para determinar la importancia de los estándares del Comité 802

del IEEE", Facultad de Informática, Universidad Autónoma de Querétaro, pág 1. Disponible: http://mena.com.mx/gonzalo/maestria/redes/Relevancia802.pdf

[40] Howard A., Robert C. B., "Álgebra Linear Contemporânea", 2a. reimpresión, Brasil, ARTMED Editora S.A.,2011, págs. 261−265.